



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

MIGRAČNÍ FRAMEWORK V SYSTÉMU SAP

MIGRATION FRAMEWORK IN SAP SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ HANÁK

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. JAROSLAV ZENDULKA, CSc.

BRNO 2018

Abstrakt

Cílem této diplomové práce je analyzovat proces migrace dat v rámci SAP systému a na základě této analýzy poté vhodně navrhnout a implementovat migrační framework, který bude využíván pro migrování dat mezi systémy SAP. Celý migrační framework i samotný proces migrace je vybudován na technologické platformě SAP NetWeaver a vyvíjen v jazyce ABAP. Funkčnost frameworku je představena pomocí případové studie, která je reprezentována vhodnou sadou dat.

Abstract

The goal of this diploma thesis is to analyze process of migration data within the scope of SAP systems. Based on this analysis suitably desing and implement migration framework that will be used for migration of data between SAP systems. Entire migration framework and migration proces is built on technological platform SAP NetWeaver and is implemented in programming language ABAP. Functionality of the framework is presented by the case study that is represented by proper data set.

Klíčová slova

migrace dat, EMIGALL, unbundling v energetice, SAP, SAP/ISU, SAP/CRM, systémy na plánování podnikových zdrojů, ABAP

Keywords

SAP data migration, EMIGALL, Utilities unbundling, SAP, SAP/ISU, SAP/CRM, Enterprise Resource Planning, ABAP

Citace

HANÁK, Tomáš. *MIGRAČNÍ FRAMEWORK V SYSTÉMU SAP*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Jaroslav Zendulka, CSc.

MIGRAČNÍ FRAMEWORK V SYSTÉMU SAP

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Ing. Jaroslava Zendulky, CSc. Další informace mi poskytli Mgr. Josef Veselý a firma CGI IT s.r.o. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Hanák
13. května 2018

Poděkování

Rád bych poděkoval svému vedoucímu doc. Ing. Jaroslavu Zendulkovi, CSc., za vedení diplomové práce po formální stránce a dále také Ing. Ondřeji Nepomuckému a Mgr. Josefu Veselému za odbornou pomoc.

Obsah

1	Úvod	3
2	Softwarové systémy SAP	5
2.1	Architektura řešení SAP/ISU a SAP/CRM	6
2.1.1	Databázový server	7
2.1.2	Aplikační server	7
2.1.3	Prezentační vrstva	7
2.2	Vlastnosti produktů SAP/ISU a SAP/CRM	8
2.2.1	Customizing řešení	9
2.2.2	Standardní metody rozšíření systému	9
2.2.3	Zákaznický vývoj	9
2.2.4	Změna standardu	9
2.3	Datový model systému SAP/ISU a SAP/CRM	10
3	Migrace dat do systému SAP	12
3.1	Postupy migrace dat do systému SAP	12
3.1.1	SAP EMIGALL	12
3.1.2	SAP LSMW	13
3.1.3	CRM Iniciální načtení	14
3.1.4	Migrace zákaznických dat přes rozhraní BAPI funkcí	14
3.2	Migrační framework	14
4	Případová studie – restrukturalizace (Unbundling)	16
4.1	Migrace na „unbundlované“ řešení SAP systémů	16
4.2	Definice migračních cest a objektů v rámci systému	18
4.3	Návrh modelu migračních objektů a migračních postupů	19
4.4	Strategie migrace dat	21
5	Návrh migračního frameworku	23
5.1	Cílový systém	23
5.1.1	EMIGALL v rámci frameworku	23
5.1.2	Návrh obsluhy migračních objektů	23
5.1.3	Grafický návrh obslužného reportu	25
5.1.4	Reprezentace migračních objektů	25
5.1.5	Způsob řízení migračních objektů	27
5.1.6	Databázové schéma frameworku	29
5.2	Zdrojový systém	32
5.2.1	Komunikace mezi systémy	32

5.2.2	Extrakční proces	33
5.2.3	Pomocná databáze na straně zdroje	36
5.3	Návrh kompletního postupu migrace	39
6	Implementace frameworku	41
6.1	Programovací jazyk ABAP	41
6.2	Pracovní nástroje a vývoj	42
6.2.1	SAP GUI a ABAP Workbench	42
6.2.2	Eclipse IDE s ABAP vývojovým nástrojem	44
6.3	Realizace navržené architektury v jazyce ABAP	45
6.3.1	RFC funkce	45
6.3.2	Rozložení implementace do paketů	46
6.3.3	Cílový systém	46
6.3.4	Zdrojový systém	48
6.4	Testování výsledného frameworku	50
6.4.1	Omezení vzorku dat	50
6.4.2	Další implementace spojené s testovacími běhy	51
6.4.3	Vyhodnocení	52
7	Závěr	53
	Literatura	54
A	Obash CD	56

Kapitola 1

Úvod

Téměř všechny velké firmy na světě dnes pracují s informacemi, a to ať už jsou právě předmětem jejich podnikání, nebo slouží pouze jako podpora v jejich podnikání. Všechny tyto informace, kdy je vidět značný stoupající trend v množství uchovávaných informací, je nutné bezpečně a efektivně uložit. Pro uložení všech těchto dat se používá databáze. Zpravidla je tento systém obohacen také softwarovými prostředky, které slouží pro manipulaci a přístup k uloženým datům. Takto definovaný software se v češtině označuje jako systém řízení báze dat (SŘBD). Obecně se dnes pod pojmem databáze rozumí samotná databáze s daty včetně SŘBD. Databáze jako taková se stala základním stavebním kamenem pro softwarový produkt Enterprise Resource Planning (ERP) [15], v češtině též plánování podnikových zdrojů. Jedná se o softwarový produkt, který má za účel pomoci informačních technologií pomáhat při řízení klíčových podnikových procesů. Proces má několik základních definicí, obecná může znít například takto: „Posloupnost činností, které se musí pravidelně vykonávat, aby bylo dosaženo požadovaného efektu.“ Do definice je také často zapojen zákazník, který je v oblasti moderního pojetí procesní organizace hlavní jednotkou. Právě zmíněným požadovaným efektem je myšlena vytvořená přidaná hodnota, za kterou je zákazník ochoten zaplatit. Cílem ERP systému je tedy sbírat, ukládat, spravovat a interpretovat data z explicitně popsaných podnikových procesů.

Jednou z firem, která se zabývá vývojem zmíněných systémů, je firma SAP. Tato firma vyvinula produkt SAP ERP (starší označení SAP R/3) potažmo jeho odvětvovou komponentu SAP/ISU, která je jedním z klíčových pojmů pro tuto práci. Komponenta SAP/ISU je odvětvově specifické rozšíření SAP ERP systému, navržené pro společnosti podnikající v utilitním odvětví. Označení utilitní společnost, či utilitní průmysl je převzato z angličtiny, kdy pod tímto pojmem rozumíme společnosti, jejichž předmětem podnikání je poskytování služeb v oblasti komodit (výroba, distribuce a prodej elektřiny, plynu, vody apod.). Systém SAP/ISU je v této oblasti velice rozšířený, neboť přesně vyhovuje potřebám velkých společností působících v tomto odvětví. Kromě systému na podporu samotných podniků byl vytvořen také produkt sloužící k optimalizaci a zefektivnění vztahu k zákazníkovi. Tento produkt je označován jako SAP/CRM. Pojem CRM značí Customer Relationship Management, tedy česky řízení vztahů se zákazníky. Tento systém zajišťuje informační podporu pro všechny zákaznický orientované oblasti od marketingu až po prodej. Pro ERP systém rozšířený o komponenty SAP/ISU a SAP/CRM se občas používá také souhrnné označení „zákaznický systém“ [11, 15, 20].

Při provozu zákaznických systémů vzniká velká sada vzájemně provázaných dat, často obchodně citlivých a klíčových k provozování činnosti těchto společností. V rámci životního cyklu podnikových systémů však čas od času (řádově jednou za 10 let) dochází k poža-

stavku na jejich obnovu či reimplementaci vyvolanou technologickými nebo jinými požadavky (např. legislativními). Součástí implementace těchto systémů je také zajištění jejich iniciačního naplnění daty z původních systémů. Tomuto procesu se obecně říká migrace dat. Při migraci dochází k přenosu předem přesně definovaných dat. Tato práce se zabývá také definicí těchto dat a podmínkami, které jsou na migrovaná data kladeny, jako je například posloupnost migrovaných dat apod. Pro podporu procesu migrace především zákaznických dat bude využit zákaznický migrační framework, který zajistí provedení jednotlivých kroků migrace zákaznických objektů, ale také kontrolu správné posloupnosti jednotlivých kroků dle definované strategie migrace. Samotná funkčnost tohoto migračního frameworku bude představena pomocí případové studie.

Cílovým systémem migrace dat bude kromě systému SAP/ISU také systém SAP/CRM. Stejně, jako se sbírají, ukládají a zpracovávají informace v SAP/ISU, je tomu tak i v produktu SAP/CRM, avšak jak se ukáže v dalších kapitolách, jisté rozdíly a omezení vyplývající z odlišností v datovém modelu i technologii řešení jiný systém přinese. Znamená to pak, že není možné předpokládat všechny implementace v obou systémech, a to ať už se to týká celých jednotlivých funkcionalit, nebo dokonce prvků na technické úrovni, tedy úrovni objektů tzv. ABAP workbench (zdrojové kódy programů) a data dictionary (definice datových domén a prvků).

Text této práce je rozdělen do následujících kapitol. Kapitola 2 popisuje obecné informace ohledně firmy SAP a jejich produktů. Je v ní rozebrán způsob dalšího rozšíření těchto produktů nad rámec dodávaného standardu a následně je nastíněn datový model SAP/ISU a SAP/CRM. V kapitole 3 jsou popsány jednotlivé způsoby, kterými lze migrovat data mezi systémy SAP. Aby bylo možné si tuto problematiku lépe představit, je uvedena reálná případová studie popisující rozdělení SAP systému na dva nové z důvodu změny legislativy. Tato motivační případová studie je popsána v kapitole 4. Kapitola 5 již představuje návrh migračního frameworku na základě analýzy z předchozích kapitol. V následující kapitole 6 je pak popsán způsob implementace tohoto frameworku v jazyce ABAP. Tato závěrečná část je zaměřena na specifika a možnosti vývoje v jazyce ABAP a popisuje způsob testování implementovaného frameworku.

Kapitola 2

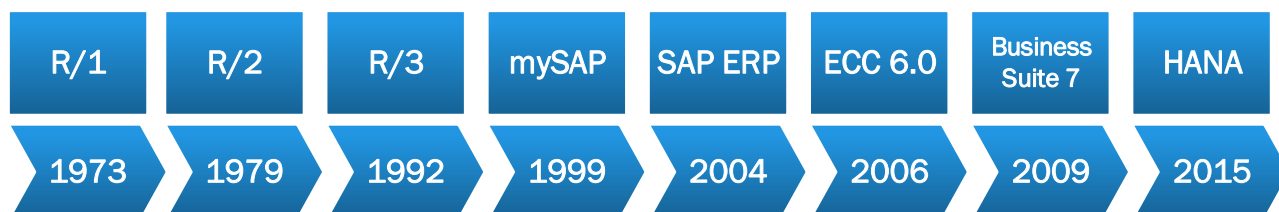
Softwarové systémy SAP

Firma SAP (z německého Software, Anwendungen und Produkte in der Datenverarbeitung), která byla založena v roce 1972, je německou IT společností, působící na trhu s podnikovými aplikacemi, a je také jednou z celosvětově největších softwarových společností. V portfoliu nabízených produktů firmou SAP najdeme software určený na řízení podniku, řešení pro datové sklady a business intelligence, software pro malé a střední podniky, platformy pro vývoj webových a standardních aplikací, software pro integraci jednotlivých počítačových systémů, ale také moderní řešení pro cloud computing apod. [11].

Prvním softwarem této firmy se stal standardní software pro oblast finančního účetnictví, který vytvořil základ systému R/1, kde písmeno R v označení značí první písmeno z Real Time-Datenverarbeitung, tedy zpracování dat v reálném čase. Z toho systému se postupem času vyvinul systém SAP R/2, který lze považovat za první ERP systém firmy SAP. V roce 1992 uvedla firma na trh produkt SAP R/3, systém zcela přepracovaný, který je založený na architektuře klient-server, a který využívá relačních databází [20].

Klíčovým produktem společnosti SAP je řešení pro plánování podnikových zdrojů (SAP ERP), které vzniklo jako následník systému R/3. Tento systém je lokalizován do více než 45 jazyků a v současné době ho využívá více než 50 000 zákazníků. Řešení navíc obsahuje celou řadu odvětvově specifických rozšíření. Odvětvová řešení společnosti SAP se historicky dělí do čtyř oblastí: Výroba, Služby, Finanční sektor a Veřejný sektor. V praxi však existuje dohromady 24 různých skupin odvětví. Například Automobilový průmysl (Automotive), Univerzity a výzkum (Higher Education & Research), Těžba (Mining) atd. Odvětvové řešení pro oblast utilit nese označení SAP/ISU (SAP Industry Solution for Utilities) a je obvykle využíváno společně s produktem SAP/CRM (Customer Relationship Management). Od verze SAP ECC 6.0 je SAP/ISU nativní součástí SAP ERP produktu (ve starších verzích byla nezbytná jeho doinstalace) [9].

Kromě samotného SAP ERP je společností SAP vyvinuto několik dalších podnikových aplikací či komponent. SAP ERP spolu s těmito dalšími komponentami tvoří produkt těsně integrovaných podnikových řešení, který se nazývá SAP Business Suite. V neposlední řadě je třeba zmínit produkt SAP HANA, který je označován za revoluční produkt a je to také jeden z nejnovějších produktů firmy. Na obrázku 2.1 je znázorněn vývoj produktů SAP vzhledem k systému ERP.



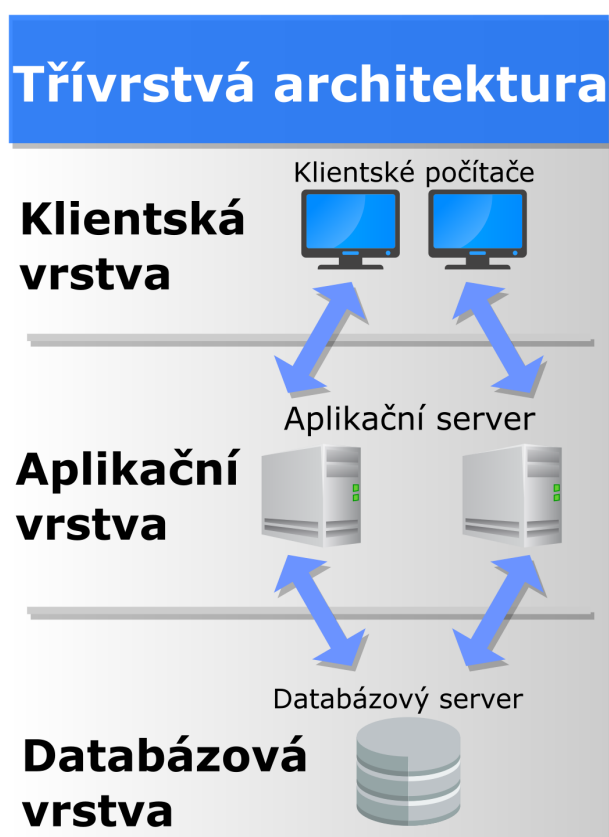
Obrázek 2.1: Vývoj produktů SAP.

2.1 Architektura řešení SAP/ISU a SAP/CRM

Architektura řešení SAP/ISU (jakožto součást SAP ERP) i SAP/CRM je vybudována na principech třívrstvé klient-server architektury a obsahuje následující komponenty [5]:

- databázový server,
- aplikační server,
- prezentační vrstvu na klientské straně.

Takováto architektura je znázorněna na následujícím obrázku 2.2.



Obrázek 2.2: Třívrstvá architektura systému [5].

2.1.1 Databázový server

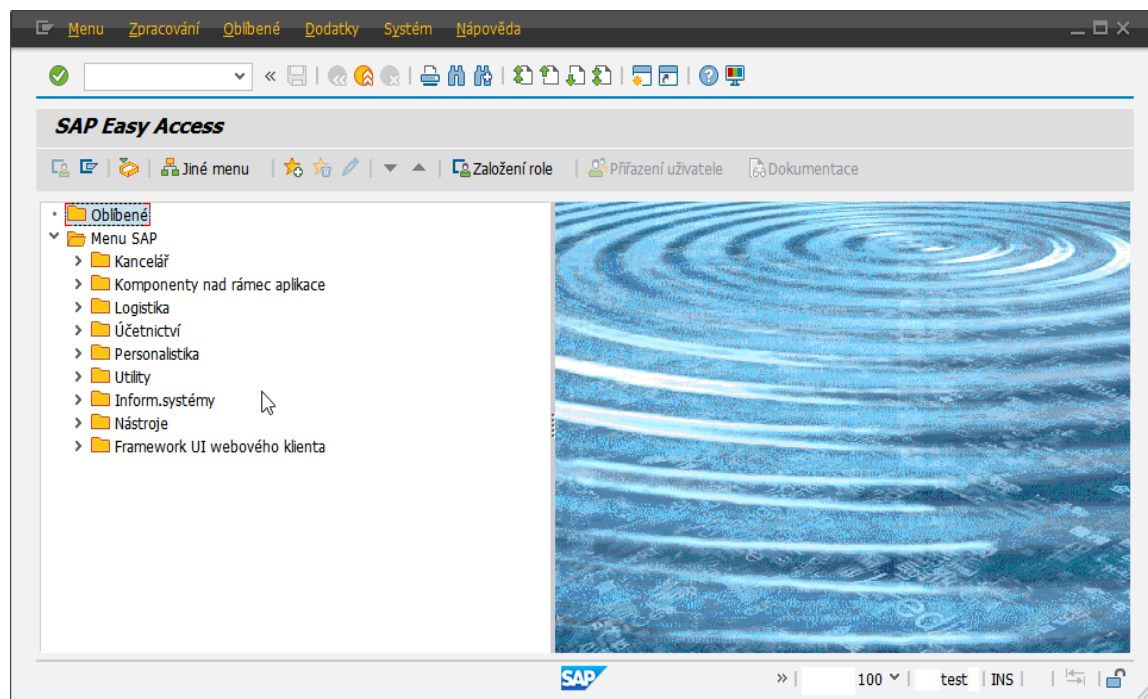
SAP ERP řešení (včetně SAP/ISU a SAP/CRM) lze provozovat na řadě běžných databázových produktů, které nejsou dodávány přímo společností SAP. Nejběžnějšími produkty jsou databáze Oracle, DB2 či Microsoft SQL Server. Relativně novou možností je databáze SAP HANA. SAP HANA je inovační aplikační a databázová platforma firmy SAP. Jedná se o in-memory, sloupcově uspořádanou (column-oriented) databázovou platformu, umožňující rychlé zpracování velkých objemů dat a jejich analýzu v reálném čase. V portfoliu produktů SAP se s databází SAP HANA výhledově počítá jako s jediným podporovaným ERP řešením – tato verze ERP produktu nese označení SAP S/4 HANA.

2.1.2 Aplikační server

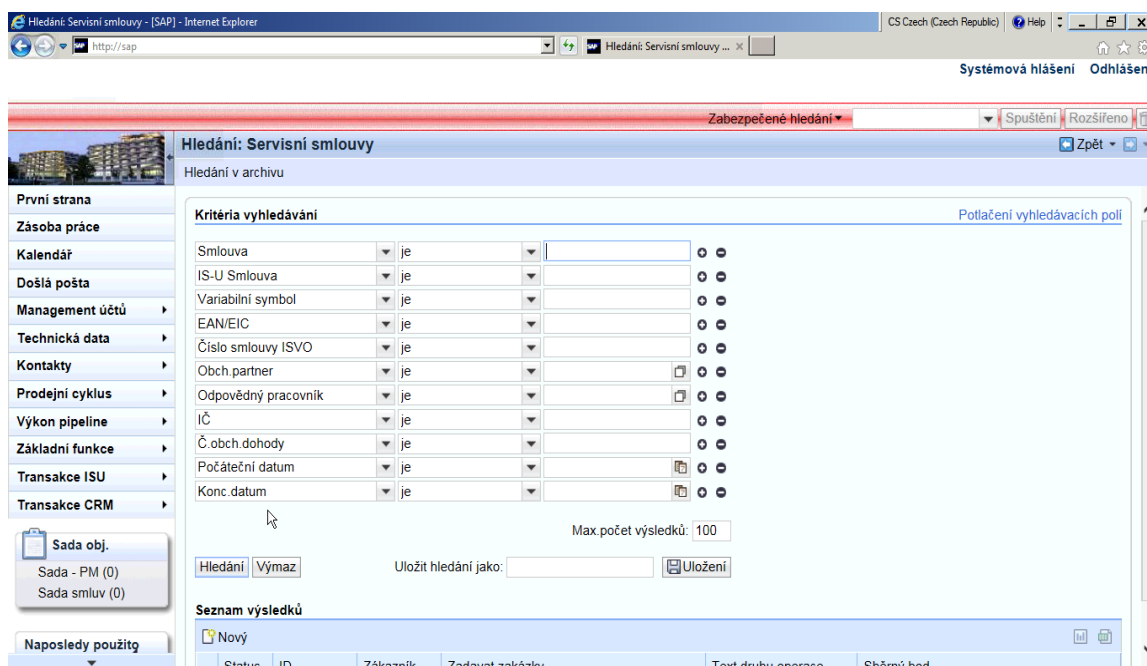
Veškerá business logika zákaznického systému je implementována na straně aplikačního serveru prostřednictvím programovacího jazyka SAP ABAP. Vlastní spouštění programů je zabezpečeno SAP běhovým prostředím, pro které se používá označení kernel.

2.1.3 Prezentační vrstva

Prezentační vrstva slouží ke zprostředkování informací uživatelům systému. Z pohledu produktů SAP/ISU a SAP/CRM jsou jako prezentační vrstva využívány zejména SAP GUI a SAP WebUI – SAP GUI reprezentuje SAP ERP tlustého klienta. Jedná se o software (aplikaci), který lze provozovat na OS Windows i Apple Macintosh. Přístup do systému SAP/ISU je běžně realizován téměř výhradně přes SAP GUI klienta. V rámci SAP GUI se systém SAP obsluhuje spouštěním tzv. transakcí. Ukázka domovské obrazovky SAP GUI lze vidět na obrázku 2.3.



Obrázek 2.3: Ukázka SAP GUI.



Obrázek 2.4: Ukázka SAP WebUI.

SAP WebUI – na rozdíl od SAP/ISU je pro obsluhu systému SAP/CRM většinou využíváno prostředí tenkého klienta (webového prohlížeče). Jedna z obrazovek SAP WebUI je ukázána na obrázku 2.4.

Od roku 2013 je možno uživatelské rozhraní společnosti SAP realizovat také s využitím nového jazyka SAP, známého pod označením SAP Fiori. SAP Fiori definuje typy aplikací, jejich vzhled a chování. Fiori aplikace jsou uzpůsobeny pro prohlížení na mobilních zařízeních (telefon/tablet), nutno ovšem dodat, že v kontextu systému SAP/ISU a SAP/CRM není toto rozhraní zatím příliš rozšířené.

2.2 Vlastnosti produktů SAP/ISU a SAP/CRM

Systém SAP je považován za otevřenou platformu. Jedná se tedy o to, že dodávaný systém SAP může být modifikován zákazníkem podle jeho potřeb. Přes opravdu širokou nabídku modifikovaných standardních řešení, jako je například právě odvětvové řešení pro společnosti působící na trhu s elektřinou SAP/ISU, není možné pokrýt všechny specifické potřeby rozličných zákazníků. Z tohoto důvodu je v systému několik možností, jak rozšířit, či modifikovat standardní předdefinované funkcionality.

Tyto možnosti lze rozdělit na následující čtyři kategorie rozšíření [20]:

- individuální nastavení systému (tzv. Customizing řešení),
- standardní metody rozšíření systému (user-exit, BADI, enhancement pointy),
- zákaznický vývoj,
- změna standardu.

2.2.1 Customizing řešení

Na rozdíl od malých ERP řešení umožňuje řešení SAP možnost přizpůsobení systému na úrovni změny parametrů. Tato customizace řešení bývá významnou součástí implementace SAP řešení a je nazývána customizací systému. Je obsluhována přes stromovou adresářovou strukturu, která je přístupná přes rozhraní SAP GUI (transakce SPRO¹) a zahrnuje možnost nastavení parametrů v řádu tisíců řídicích tabulek s nastavením.

2.2.2 Standardní metody rozšíření systému

Pro případy, kdy není možno zákaznické požadavky na cílové řešení vyřešit na úrovni nastavení systému (customizace), existují v rámci řešení SAP standardní metody rozšíření systému. V principu se jedná o předpřipravená místa ve zdrojovém kódu s definovaným rozhraním, která umožňují úpravu standardní funkcionality formou zákaznického vývoje. Existují následující dva typy standardního rozšíření systému SAP.

- **User-Exit** – tradiční způsob rozšíření systému, kdy do standardního SAP kódu (funkčních modulů, reportů, definic datových struktur) byla definována místa v podobě těchto exitů, které lze implementovat bez nutnosti zasahovat do původního kódu – existuje mnoho typů exitů pro rozšíření různých objektů SAP workbench a data dictionary (tyto pojmy jsou podrobně vysvětleny v pozdějších kapitolách): screen exity, exity struktur realizované formou zákaznických „includů“, funkční exity.
- **BADI** – novější objektově orientovaný koncept známý jako SAP Business Addins, rozšíření jsou realizována formou reimplementace předpřipravených metod definovaných tříd.

2.2.3 Zákaznický vývoj

Tam, kde pouhá úprava standardní funkcionality je nedostačující a je potřeba implementovat zcela novou funkcionalitu, se přistupuje k tzv. zákaznickému vývoji [21]. Pro tyto potřeby je firmou SAP v jejich systémech vyhrazen zákaznický jmenný prostor, který je určen právě k vývoji nových funkcionalit nad rámec standardních možností. Tento jmenný prostor spočívá ve vyhrazení písmen „Z“ a „Y“, která je nutné použít při vývoji vlastních funkcionalit. V praxi to pak znamená to, že je vyvíjena například zákaznická nadstandardní funkcionalita pro vypsání známého „Hello World!“. Samotný program je potřeba pojmenovat například Z_HELLOWORLD. Tento postup, který je vyžadován, má jednoduchý důvod. V případě, že by byl zákaznický vývoj pojmenován například pouze HELLOWORLD, SAP nezaručuje, že se ve své aktualizaci nerozhodne vyvinout svoji funkcionalitu pojmenovanou stejně. V tomto případě by byl zákaznický vývoj nenávratně ztracen a přehraán standardní funkcionalitou. Z toho plyne, že všechny zákaznické vývoje musí podléhat konvenci pojmenování „Z“, či „Y“, a to ať je vyvíjena například nová třída, funkční modul nebo cokoliv jiného.

2.2.4 Změna standardu

Úpravu standardního chování systému je možno realizovat i formou změny standardních SAP objektů. Takto změněné objekty musí být vždy k dané instalaci SAP registrovány

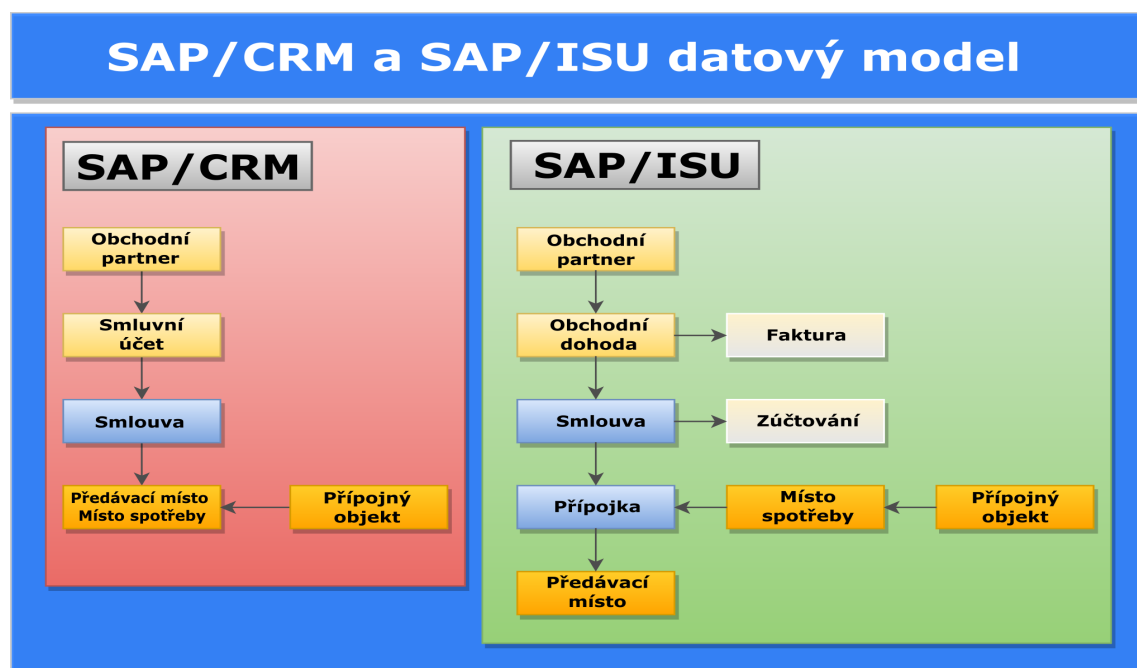
¹ Transakce je standardní pojem SAP a reprezentuje program spustitelný z příkazové řádky SAP GUI.

na SAP servisním portále (webová stránka service.sap.com), kde je také vygenerován klíč nezbytný k registraci příslušné reparatury. Změna standardních objektů má dopady na záruku poskytovanou společností SAP a na funkčnost dodaného řešení (SAP neručí za správnost chování změněných objektů). Tato změna také přináší potenciální problémy při periodických aktualizacích řešení, kdy SAP nezaručuje jejich funkčnost při přechodu na vyšší verzi (na rozdíl od výše uvedených zpětně kompatibilních metod vývoje). Pro potřeby modifikace standardního SAP kódu je v SAP k dispozici odpovídající nástroj, tzv. modifikační asistent. Tento asistent právě usnadňuje přechod na vyšší verze tím, že při aktualizaci dokáže změněná místa najít, vypsat a navrhnout způsob jejich začlenění do nového zdrojového kódu tam, kde je to možné. Z výše uvedeného popisu je zřejmé, že změna standardu je až poslední možností a jejich počet by v rámci implementace řešení měl být minimalizován.

2.3 Datový model systému SAP/ISU a SAP/CRM

Datový model zákaznického systému SAP/ISU a SAP/CRM je na technické úrovni velmi komplexní a zahrnuje stovky až tisíce databázových tabulek. Z tohoto důvodu se přistupuje ke zjednodušenému podnikovému popisu jednotlivých objektů, které již reprezentují skutečné objekty v reálném světě. Data v zákaznickém systému se dělí na 2 základní kategorie.

- **Kmenová (master) data** – jedná se typicky o časově relativně stálá (neměnná) data reprezentující kmenové záznamy zákazníků a jejich odběrných míst.
- **Pohybová (transakční) data** – dynamicky rychle vznikající a měnící se data, v kontextu zákaznického systému se jedná např. o zúčtovací doklady, faktury, data o přístrojích (elektroměrech), odečtech spotřeby apod.



Obrázek 2.5: Datový model SAP systémů [3].

Na obrázku 2.5 je uveden výrazně zjednodušený model zákaznického systému SAP/ISU a SAP/CRM. Žlutou barvou (partner a smluvní účet) jsou znázorněna obchodní kmenová data. Technická kmenová data jsou značena barvou oranžovou. Modře značené prvky jsou pak data, která slouží k propojení předchozích dvou skupin a poslední skupina (faktura a zúčtování) jsou již zmíněná pohybová data. Vazby značí posloupnost závislostí. Například nelze mít fakturu, pokud nemáme definovanou obchodní dohodu atd.

Kapitola 3

Migrace dat do systému SAP

Součástí nasazení nového SAP/ISU řešení (též označováno jako implementace nového systému) bývá ve většině případů i požadavek na zajištění iniciálního naplnění systému daty. Požadavek na migraci dat bývá velmi komplexní a jedná se o jeden z klíčových předpokladů úspěšné realizace projektu implementace systému SAP u zákazníka. Z důvodu složitosti datového modelu v systému SAP se migrace dat nerealizuje na úrovni databázových tabulek, ale na úrovni SAP Business datových objektů (viz datový model SAP/ISU a SAP/CRM). V takovémto případě je v systému nutné dodržet přesný postup, resp. pořadí migrovaných objektů. Objekty jsou totiž zpravidla v systémech na sobě závislé a k migraci některých migračních objektů může dojít až po úspěšné migraci objektů jiných.

3.1 Postupy migrace dat do systému SAP

Migrování dat do systému SAP je umožněno pomocí několika standardních nástrojů.

3.1.1 SAP EMIGALL

SAP EMIGALL [4, 17] je standardní SAP migrační „pracovní nástroj“¹ poskytující otevřené rozhraní pro přenos kmenových a pohybových dat do systému SAP/ISU. Je určen pro migraci dat do SAP/ISU pro různé typy zdrojových systémů bez ohledu na technologii či datový model.

Migrace dat do SAP/ISU je postavena na migraci podnikových datových objektů. Vzhledem ke složitosti a komplexitě datového modelu SAP/ISU není migrace dat na úrovni databázových tabulek možná. EMIGALL z tohoto důvodu využívá přístup pomocí takzvaných migračních objektů, který umožňuje objektově orientovaný import dat. Import dat využívá technologii direct-input, přímého volání SAP BAPI (Business Application Programming Interface) – ta oproti importu dat přes BDC (přes GUI, tedy Batch Input, který je popsán dále) přináší výhodu z hlediska výkonu.

¹ Z anglického workbench, kde původně se pod pojmem v SAP prostřední rozuměl soubor nástrojů určených pro vývoj aplikací v jazyce ABAP, ale později se název začal používat i pro další nástroje v SAP.

Využití nástroje standardního ISU migračního nástroje (EMIGALL) přináší oproti technickému (databázovému) přístupu k migraci následující výhody.

- Nezávislost na datovém modelu zdrojového systému (konverze datových modelů je provedena na úrovni extrakčních nástrojů dle požadavků zákazníka definovaných ve fázi cílového konceptu).
- Podnikově orientovaný přístup (na místo technického na úrovni tabulek) umožňující zapojení klíčových uživatelů do designu migrace dat.
- Garantuje provedení aplikačních kontrol a kontrol vzájemné konzistence migrovaných dat.
- Direct-input technologie poskytuje optimální výkonnost pro zkrácení doby odstávky (pro migraci velkých objemů dat se dále předpokládá využití paralelizace procesu importu dat).
- Pro import dat jsou využity již existující standardní metody a funkce.
- Nástroj využívá obecné migrační struktury a umožňuje jejich snadné rozšíření za účelem migrace zákaznických dat standardních objektů.
- Řízení chyb (tzv. Error handling) a poskytování protokolů s výsledky migrace jednotlivých objektů.

Prostřednictvím nástroje EMIGMASSRUN, pak EMIGALL umožňuje distribuované paralelní zpracování migračních objektů, u kterých se předpokládá dlouhá doba běhu za účelem zrychlení procesu migrace dat.

3.1.2 SAP LSMW

SAP LSMW (ze zkratky Legacy System Migration Workbench) je softwarový nástroj firmy SAP, který podporuje migrování dat z jiných systémů, než je SAP, právě do systému SAP. Jedná se o nástroj postavený na logice systému R/3. Pomocí tohoto nástroje lze data z ne-SAP systému přenášet jednorázově, ale případně také opakovaně. S jistými omezeními lze tento nástroj používat také k migrování dat ze systému SAP do systému SAP. Obecně se však používá tento nástroj k migrování dat z historických systémů (často převzato z angličtiny též jako „legacy“ systém) [18].

SAP LSMW zajišťuje následující funkcionalitu:

- načítání dat z tabulkových procesorů nebo ze sekvenčních souborů,
- převádění dat z originálního formátu do formátu cílového SAP systému,
- importování dat pomocí standardního SAP rozhraní (IDoc inbound processing, Batch Input, Direct Input).

IDoc Celým názvem Intermediate Document je SAP objekt, který je schopen uchovávat data v rámci aplikační transakce z jednoho systému do druhého ve formě elektronické zprávy. Zpracování IDoc dokumentu systémem SAP, tedy zpracování příchozího IDoc dokumentu z jiného systému se nazývá IDoc inbound processing.

Batch input Tímto pojmem označujeme dva pojmy zároveň. Zaprvé se tím rozumí standardní ERP rozhraní a zadruhé procedura pro migrování dat. Tato zavedená technologie umožňuje naplnit dialogové transakce dodanými daty.

Direct input Jelikož propustnost skrze Batch input není vždy dostačující, je zapotřebí napsat programy pro přenos některých datových objektů, které zapíší objekty přímo, tedy přímo do databáze SAP ERP systému. Těto metodě se říká Direct input.

3.1.3 CRM Iniciální načtení

Systém CRM lze naplnit pomocí tzv. Iniciálního načtení (anglicky Initial Load) ze systému SAP/ISU. Je zde tedy nutné nejprve migrovat data do nového systému ISU, kde lze následně použít Iniciální načtení. Pod tímto pojmem se rozumí standardní SAP transakce. Pojmem transakce se v SAP rozumí jakási obálka, která dělá program spustitelný rovnou z příkazové řádky SAP GUI, kdy není nutné nalézt konkrétní implementaci a spouštět ji napřímo. Tato standardní SAP transakce pak umožňuje prvotní načtení dat ze systému SAP ERP, tedy všechny business a uživatelská data, a vloží je do CRM systému. Tímto způsobem lze nový CRM systém naplnit počátečními daty [10].

3.1.4 Migrace zákaznických dat přes rozhraní BAPI funkcí

BAPI (zkratka z Business Application Programming Interface) reprezentuje precizně definované rozhraní, které umožňuje přístup k procesům a datům v podnikových aplikačních systémech, jako je SAP R/3, tedy i ISU a CRM. BAPI jsou definovány jako API metody pro SAP data business objekty. Původně bylo BAPI navrženo pro přístup k SAP ERP z vnějšího prostředí, kdy datové objekty většinou četly a zapisovaly jednotlivé BAPI. Postupem času se však vyvinuly v možnost přenášet data do databáze systému SAP ERP v průběhu migrace dat [23].

3.2 Migrační framework

Jak je již uvedeno v předcházejících podkapitolách, existuje několik postupů migrace dat do zákaznických systémů. V rámci velkých projektů je však zpravidla nezbytné tyto postupy kombinovat a zajistit správnost postupu migrace pro objekty migrované výše uvedenými technologiemi. Spolu s požadavkem na automatizaci procesu extrakce a importu dat, automatizaci obsluhy, poskytnutí souhrnných reportů k prokázání kompletnosti a kvality namigrovaných dat, byl jeden z hlavních důvodů vzniku komplexního zákaznického migračního frameworku, pro podporu migrace dat do rozděleného řešení utilitní společnosti (viz kapitola 4).

Hlavní požadavky na migrační framework:

- objektově orientovaný přístup (přístup založený na tzv. migračních objektech) s možností customizace migračních objektů,
- uživatelská administrace migrace pomocí migračního kokpitu se zahrnutím objektů migrovaných přes EMIGALL,
- administrace filtrů pro migraci dat pro podporu testování migrace,
- management migračních klíčů (ID objektů ze stávajícího a cílového řešení) a stavů migrovaných objektů,
- pomigrační kontroly a migrační log k prokázání kvality a kompletnosti namigrovaných dat.

Kapitola 4

Případová studie – restrukturalizace (Unbundling)

Přibližně od roku 2003 dochází v Evropské unii k procesu liberalizace trhu s elektřinou a plynem. Jedná se o postupný proces, jehož součástí je také tzv. unbundling¹ velkých energetických společností. Tento pojem můžeme přeložit jako oddělení ve smyslu rozkladu celku na menší diverzifikované části. Přesně to je totiž unbundling, ve kterém dochází k trvalému rozdělení vertikálně integrovaných energetických skupin a osamostatnění těch činností, které jsou přirozeným monopolem, od činností ostatních. Vertikálně integrovanou energetickou společností označujeme společnost, která působí společně v oblasti přenosu, distribuce a prodeje, případně i těžby a výroby (v českém kontextu se jedná o společnosti typu ČEZ, E.ON či Innogy – dříve RWE).

Od roku 2006 probíhá restrukturalizační proces také v České republice, a to v rámci reakce na evropskou legislativu, která se snaží o tzv. liberalizaci energetiky v Evropské unii, jež má vytvořit jednotné celoevropské konkurenční prostředí nebo také zaručit spolehlivost dodávek a transparentnost trhu s elektřinou v rámci EU. Liberalizace trhu, respektive již zmíněná restrukturalizace, představuje důsledné oddělení konkurenčních segmentů. Těmito segmenty se myslí výroba a prodej versus přirozené monopoly představující přenos a distribuci energie. Toto rozdělení umožňuje pak regulaci a zajišťuje rovný přístup k sítím pro všechny společnosti, které operují právě v konkurenčních částech trhu.

Jak tedy bylo nastíněno, unbundling je postupný proces, který má probíhat na všech následujících rovinách [22]:

- **právní** – vyčlenění regulovaných činností do samostatných právních subjektů (dceřiných společností),
- **manažerské** – oddělené vedení takto vyčleněných společností,
- **finanční** – samostatné vedení účetnictví a vykazování hospodářského výsledku,
- **informační** – oddělení na úrovni informačních systémů.

4.1 Migrace na „unbundlované“ řešení SAP systémů

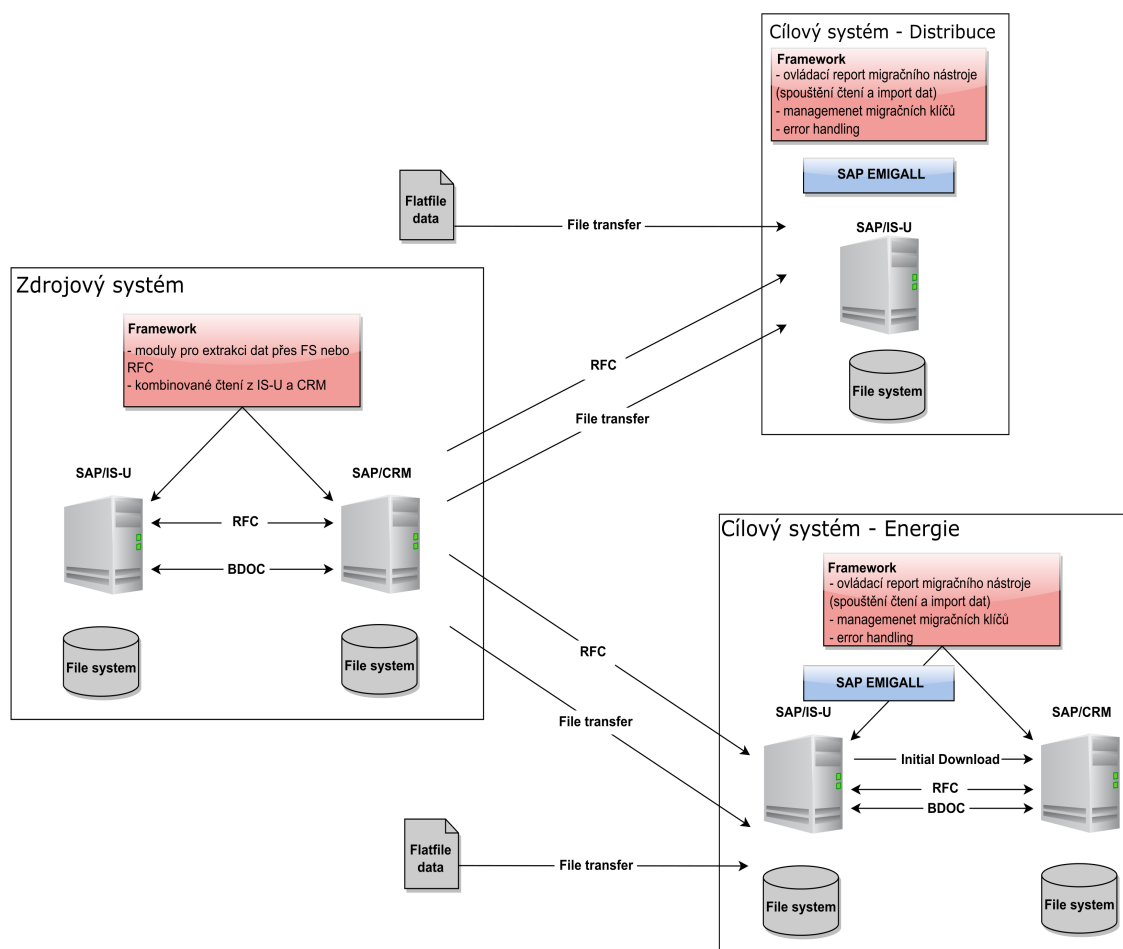
Popsaná skutečnost však vyžaduje značné úpravy v takto dotčených společnostech. Jednou z těchto úprav a zároveň i nejvýznamnější, co se týče investiční složky takové úpravy,

¹ V praxi se používá zejména výraz unbundling.

je pak právě implementace nových ERP, bez kterých je nemožné se obejít. Přesně na takovéto migraci, řekněme tedy případové studii, bude představen koncept migračního frameworku, ale také migrace jako takové. Z pohledu níže popsané případové studie je relevantní právě informační unbundling realizovaný v rámci případové studie jako přechod z jednoho společného systému společnosti poskytující prodej a distribuci elektrické energie a plynu do dvou nově implementovaných samostatných systémů. Součástí implementace je samozřejmě také zajištění iniciálního naplnění systému daty – migrace dat.

Jedná se o případovou studii, která přesně odpovídá takovému restrukturalizovanému řešení, v němž bude prováděna migrace z jednoho společného systému společnosti poskytující prodej a distribuci komodity. V tomto systému jsou veškerá data neoddělaná a sdílená. V rámci restrukturalizace bude provedena migrace do dvou oddělených zákaznických systémů, kdy první slouží pro účely konkurenční části trhu (obchodu s komoditou). Tento systém budeme dále označovat jako „energie“. Druhý (distribuční) systém, který je určen pro monopolistickou (regulovanou) část trhu, budeme označovat jako „distribuce“.

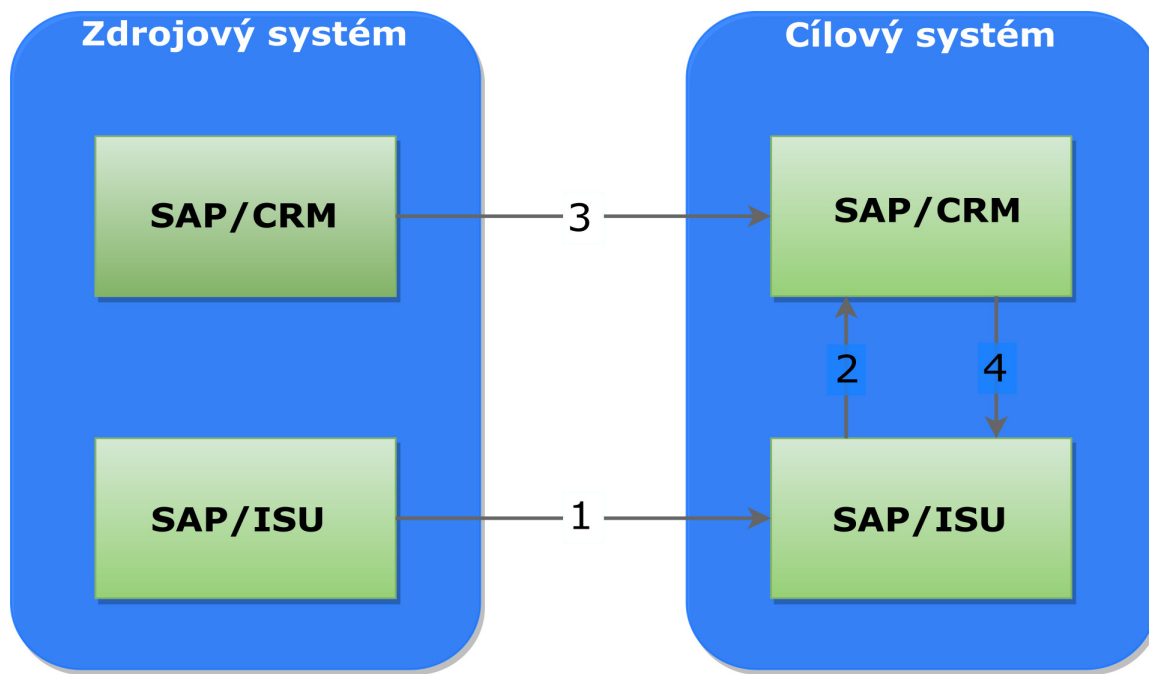
Zdrojové i cílové řešení je postaveno na produktech SAP/ISU a SAP/CRM, schéma takto popsaných migračních systémů a způsobu komunikace a migrace dat je vidět na obrázku 4.1, detailněji budou spojení popsána v pozdějších kapitolách.



Obrázek 4.1: Schéma celé migrace v rámci případové studie – Unbundling.

4.2 Definice migračních cest a objektů v rámci systému

Vzhledem k tomu, že na straně zdrojového i cílového řešení probíhá replikace dat mezi komponentami SAP/ISU a SAP/CRM, bylo v rámci studie nezbytné provést analýzu a definovat možné migrační cesty. Jejich schéma je uvedeno na následujícím obrázku:



Obrázek 4.2: Migrační cesty.

1. Migrace objektů SAP/ISU do SAP/ISU zpravidla využívá technologii EMIGALL:
 - využití pro objekty technických kmenových dat, kmenových dat obchodních partnerů, smluvních účtů a dále smluvních účtů (označovaných jako FICA objekty), tedy pohybová data.
2. Iniciální načtení objektů do SAP/CRM:
 - pro objekty kmenových dat (technická kmenová data, obchodní partner a smluvní účet) bude migrace do SAP/CRM provedena technologií iniciálního načtení dat.
3. Přímá migrace CRM objektů ze SAP/CRM do SAP/CRM:
 - objekty existující pouze v SAP/CRM.
4. Standardní propad dat ze SAP/CRM do SAP/ISU:
 - pro objekty, pro které není podporována metoda iniciálního načtení.

4.3 Návrh modelu migračních objektů a migračních postupů

Tabulka 4.1: Rozdělení migračních dat.

<u>Kmenová data</u> <ul style="list-style-type: none"> o Základní funkce <ul style="list-style-type: none"> • Poštovní regionální struktura • Politická regionální struktura o Technická kmenová data <ul style="list-style-type: none"> • Přípojný objekt • Místo spotřeby • Umístění přístroje • Odběrné místo • Předávací místo (deregulační) • Fakta odběrného místa o Obchodní kmenová data <ul style="list-style-type: none"> • Obchodní partner • Smluvní účet • ISU Smlouva 	<u>Customer service</u> <ul style="list-style-type: none"> o Smlouvy pro komodity o Nabídky o Produkty komodit o Nekomoditní smlouvy (VAS) o Kontakty
<u>Tržní komunikace</u> <ul style="list-style-type: none"> o Kmenová data <ul style="list-style-type: none"> • Sítě • Poskytovatele služeb • Tržní komunikace • Přejícné období 	<u>Správa přístrojů a měřených dat</u> <ul style="list-style-type: none"> o Správa přístrojů <ul style="list-style-type: none"> • Přístroj • Infozáznam • Profily • Odečet/Data odečtu
<u>FICA</u> <ul style="list-style-type: none"> o Závazky/pohledávky o Statistické položky o Akontace o Spisy soudního vymáhání o Externí vymáhání o Prodej pohledávek o Platby/Dávky plateb o Platební příkazy o Záruky o Doručenky o Splátkové kalendáře o Bonita 	<u>Zúčtování a fakturace</u> <ul style="list-style-type: none"> o Plány záloh <ul style="list-style-type: none"> • Platební kalendáře • Opravné doklady • Migrace DPC • Ceníky

Tabulka 4.1 obsahuje migrační objekty rozdělené do logických celků. Jedná se o skupiny migračních objektů, které odpovídají případové studii. Obecně však může být model

při použití frameworku na jiných projektech různý. Za zmínku stojí skupina pojmenovaná FICA. FICA (zkratka z Financial Accounting – Contract Accounts Receivables and Payables) je finanční modul, který se používá zejména v SAP systémech zaměřených na průmysl. Skupina FICA tedy obsahuje objekty odpovídající tomuto modulu.

Na základě výše uvedených požadavků na rozsah migrovaných dat byl proveden návrh seznamu migračních objektů následujícím způsobem:

Tabulka 4.2: Přehled všech migračních objektů.

Business objekt	Migrační objekt (zkratka)	Migrační postup
Místo	ADRCITY	EMIGALL
PSC	ADRPSTCODE	EMIGALL
Ulice	ADRSTREET	EMIGALL
Politická regionální struktura		TABLE
Přípojný objekt	CONNOBJ	EMIGALL
Přípoj. objekt poznámky	ZCON_NOTE	EMIGALL
Místo spotřeby	PREMISE	EMIGALL
Místo spotřeby	ZPREM_NOTE	EMIGALL
Předávací místo	POD	EMIGALL
Předávací místo	PODSERVICE	EMIGALL
Odběrné místo	HYP_INSTLN	EMIGALL
Fakta odběrného místa	FACTS	EMIGALL
Obchodní partner	PARTNER	EMIGALL
Obchodní partner	PART_REL	EMIGALL
Obchodní partner	ZPART_NOTE	EMIGALL
Obchodní partner	ZINSOLV	EMIGALL
Smluvní účet	ACCOUNT	EMIGALL
Smluvní účet	ACC_NOTE	EMIGALL
Kontaktní strategie		BAPI
Smlouva	ZCONTRACT	EMIGALL
Infozáznam	DEVINFOREC	EMIGALL
Infozáznam	DEVICERATE	EMIGALL
Přístroj	ZDEVICE_TAB	EMIGALL
Přístroj	INST_MGMT	EMIGALL
Odečty	ZEMRDETAIL	EMIGALL
doklad FICA	POHL	EMIGALL
doklad FICA	STAT	EMIGALL
doklad FICA	AKONT	EMIGALL
spisy soudní vymáhání	ZSPIS	EMIGALL
externí vymáhání	ZCOLL	EMIGALL
Platby	PAYMENT	EMIGALL
Dávka plateb	ZPAY_LOT	EMIGALL
Platební příkazy	ZPYOR	EMIGALL
Záruky	SECURITY	EMIGALL
Záruky	PAY_SECU	EMIGALL
Doručenky	ZDOR	EMIGALL

pokračování na další straně

Tabulka 4.2 – pokračování z předchozí strany

Business objekt	Migrační objekt (zkratka)	Migrační postup
Historie upomínání	DUNNING	EMIGALL
Úroky	DOC_INT	EMIGALL
Splátkové kalendáře	INSTPLAN	EMIGALL
Bonita	ZBONIT	EMIGALL
Opravné položky	DOC_DOUBT	EMIGALL
Předpisy záloh	ZDOCUMENT_ZAL	EMIGALL
Předpisy záloh	ZDOCUMENT_ZAL	EMIGALL
Předpisy záloh	ZDOCUMENT_ZAL	EMIGALL
Předpisy záloh	ZNOTE_ZAL	EMIGALL
Zúčtovací doklady pro DPC	ZBILLDOC	EMIGALL
Tiskové doklady DPC	ZEBIT_EXT	EMIGALL
Aktivace RTP komponenty	ZRTP_INSTLN	EMIGALL
Ceníky	ZEPREI	EMIGALL
Síť	ZGRID	EMIGALL
Poskytovatel služeb	ZESRVPROV	EMIGALL
Obchodní partner	BUPA_MAIN	Initial load
Obchodní dohoda	BUAG_MAIN	Initial load
Přípojný objekt	SI_CONNOBJ	Initial load
Předávací místo, místo spotřeby	SI_POD	Initial load
Smlouva komodit	SI_CONTRACT	Initial load
Smlouva nekomoditní (VAS)	VAS	BAPI
Komplety	PACKAGE	TABLE
Nabídka	PROPOSAL	BAPI
Kontakt	CONTACT	BAPI

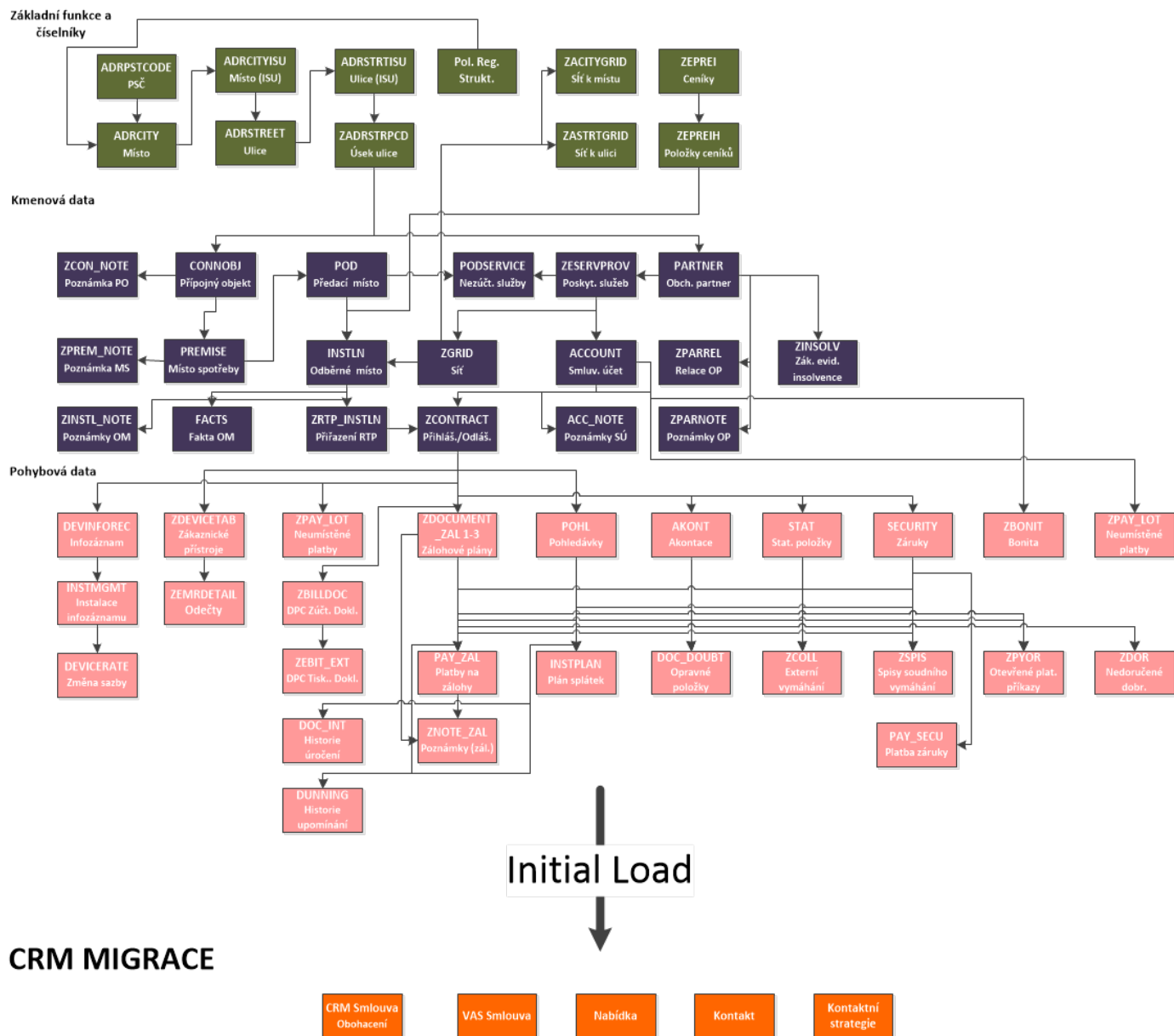
4.4 Strategie migrace dat

Vlastní migrace dat pak zahrnuje spouštění jednotlivých migračních kroků, spočívajících v extrakci a importu dat jednotlivých objektů, ale i dalších činnostech jako je např. změna nastavení a konfigurace systémů pro různé migrační fáze apod. Protože migrovaných dat je velké množství, byla stanovena strategie migrace dat, která je vyobrazena na obrázku 4.3 na následující straně.

Na tomto obrázku je vidět detailní schéma všech migračních objektů, a to jak standardních, tak i zákaznických objektů, které je třeba migrovat. Šipka vždy značí závislost, tj. $A \rightarrow B$ značí, že objekt B je závislý na objektu A. Objekt A musí být tedy migrován vždy před objektem B. Z obrázku je tedy patrné, že ze skupiny objektů Základní funkce a číselníky (zeleně značené) může být například objekt ADRCITY migrován až po úspěšném migrování objektu ADRPSTCODE a Politické regionální struktury (Pol. Reg. Strukt.).

K evidenci a kontrole správné posloupnosti jednotlivých migračních kroků pak bude sloužit právě implementovaný migrační framework, který umožní definici navržených migračních objektů, včetně nastavení a vyhodnocení jejich posloupnosti a závislostí.

ISU MIGRACE



Obrázek 4.3: Schéma všech migračních objektů včetně závislostí.

Kapitola 5

Návrh migračního frameworku

Migrační framework, jak bylo naznačeno v předchozích kapitolách, má zajišťovat vrstvu mezi již dostupnými prostředky pro migraci dat a specifickými požadavky na migraci dat konkrétními zákazníky. Dále je od migračního frameworku vyžadováno, aby zabezpečil provedení jednotlivých kroků migrace pro migrační objekty, a také, aby zajistil určitou možnost kontroly správnosti dat. Důraz je tedy kladen na seskupení jednotlivých funkcionalit do jednoho místa, odkud bude možné jednoduše tyto činnosti provádět. V následujících kapitolách je popsán právě návrh tohoto frameworku. Toto řešení předpokládá implementaci jak na systému zdrojovém, tak i na cílovém. Veškeré následující podkapitoly respektují některé zvyklosti a specifika jazyka ABAP, který je zvolen pro implementaci migračního frameworku (resp. je nutností tohoto řešení).

5.1 Cílový systém

Tento systém je určen jako systém, ve kterém bude implementováno řízení celé migrace. Je tedy nutné nejprve navrhnout reprezentaci migračních objektů v cílovém systému. Dále je zapotřebí vytvořit návrh obsluhy samotné migrace a jednotlivých úkonů. K tomuto účelu je navržen obslužný program.

5.1.1 EMIGALL v rámci frameworku

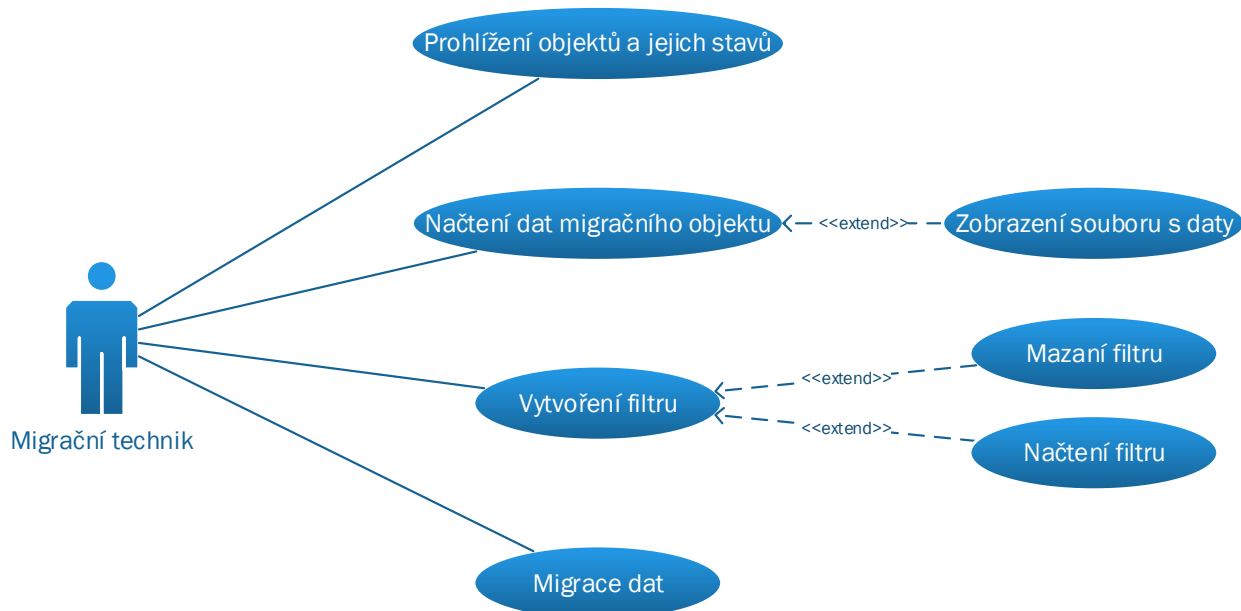
V prostředí SAP/ISU je EMIGALL (viz podkapitola 3.1.1) stěžejním nástrojem pro migrování dat. Pro následující podkapitoly postačí fakt, že EMIGALL předpokládá data migračních objektů ve formě binárního souboru s pevně danou syntaxí. Tento fakt se projeví právě při realizaci návrhu reprezentace migračních objektů.

5.1.2 Návrh obsluhy migračních objektů

Samotná migrace bude prováděna migračním technikem. Pod pojmem migrační technik si lze představit jednoho či více IT specialistů z oblasti řešení SAP s alespoň základní znalostí jazyka ABAP, kteří budou provádět migraci dat. V případě vzniku běhových chyb je nutné provést okamžitou nápravu, neboť jak bylo detailně znázorněno v kapitole 4.4, objekty jsou na sobě závislé. Byť jen jedna chyba v migraci určitého objektu může znamenat, že selže migrace stovky dalších. Představme si situaci, kdy migrujeme data pro všechny zákazníky pro konkrétní město, kde však při migraci selže nahrání jednoho poštovního směrovacího čísla, tedy PSČ. V tomto případě nebudeme mít možnost migrovat žádný dům, žádného

obchodního partnera (obecně zákazník distributora), žádnou přípojku atd., která je spojena s tímto konkrétním PSČ.

Definované požadavky na migrační framework jsou graficky zpracovány pomocí diagramu případů užití, který je vidět na následujícím obrázku:



Obrázek 5.1: Diagram případů užití.

Prohlížení objektů a jejich stavů reprezentuje základní grafickou obrazovku, která je představena v následující podkapitole. Tato akce je vyvolána uživatelem zapnutím migračního obslužného programu.

Načtení dat migračního objektu je akce spojená se samotným exportem dat konkrétních migračních objektů. Ve většině případů je požadavek na export těchto dat ve formátu binárního souboru. Zde je zakomponovaný požadavek na to, aby si právě nahraná data do cílového systému mohl migrační technik prohlédnout a zjistit tak, jestli odpovídají skutečně datům, která chtěl nahrát. Tento princip se bude hodit především při testovacích bězích, ve kterých je potřeba ověřit funkčnost extrakce dat na malém vzorku a uživatel je ještě schopen pouze prohlédnutím načtených dat určit jejich správnost.

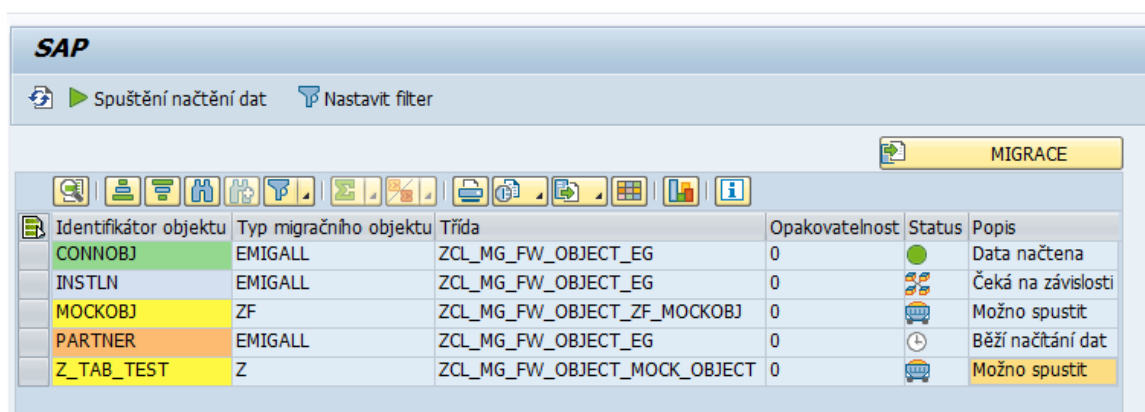
V případě migrace celého systému je v mnoha případech migrováno i přes 2 milióny konkrétních instancí migračního objektu (například 2 milióny obchodních partnerů – odběratelů elektřiny). V tomto množství je prohlížení souboru určeno převážně ke kontrole správnosti struktury načteného souboru.

Z důvodu velkého objemu dat při konečné migraci je navržen princip testovacích migračních běhů, kdy je migrace prováděna tzv. „na nečisto“ pouze na malém vzorku dat, na kterém se testuje kvalita dat a úspěšnost migrace. Tato data jsou pak vymezena pomocí uživatelského filtru. Uživatel si bude moci takovýto filtr vytvořit a uložit na pozdější použití, nebo si bude moci svůj filtr také smazat.

Poslední funkcionalitou je pak samotná migrace dat. Tímto pojmem se označuje koncový import do nového systému, kdy dojde k uložení extrahovaných dat do nového cílového systému.

5.1.3 Grafický návrh obslužného reportu

Pro řízení migrace včetně hlídání jednotlivých stavů migračních objektů bude sloužit obslužný program. Jazyk ABAP dovoluje unifikovaný přístup, který se nazývá Dynpro (Dynamický Program) [2, 14]. Jedná se o poměrně starý grafický framework, který sahá až do roku 1992, kdy byl uveden systém R/3 (viz Obrázek 2.1). Stále se však jedná u odvětvových řešení systémů SAP/ISU o nejpoužívanější způsob tvorby grafických uživatelských obrazovek. Skrze Dynpro můžeme navrhnout aplikaci s řadou UI prvků. Pro potřeby migračního frameworku je navrženo grafické uživatelské rozhraní právě pomocí tohoto grafického frameworku.



Identifikátor objektu	Typ migračního objektu	Třída	Opakovatelnost	Status	Popis
CONNOBJ	EMIGALL	ZCL_MG_FW_OBJECT_EG	0	●	Data načtena
INSTLN	EMIGALL	ZCL_MG_FW_OBJECT_EG	0	⚙️	Čeká na závislosti
MOCKOBJ	ZF	ZCL_MG_FW_OBJECT_ZF MOCKOBJ	0	🚗	Možno spustit
PARTNER	EMIGALL	ZCL_MG_FW_OBJECT_EG	0	⌚	Běží načítání dat
Z_TAB_TEST	Z	ZCL_MG_FW_OBJECT MOCK_OBJECT	0	🚗	Možno spustit

Obrázek 5.2: Návrh hlavní části migračního programu.

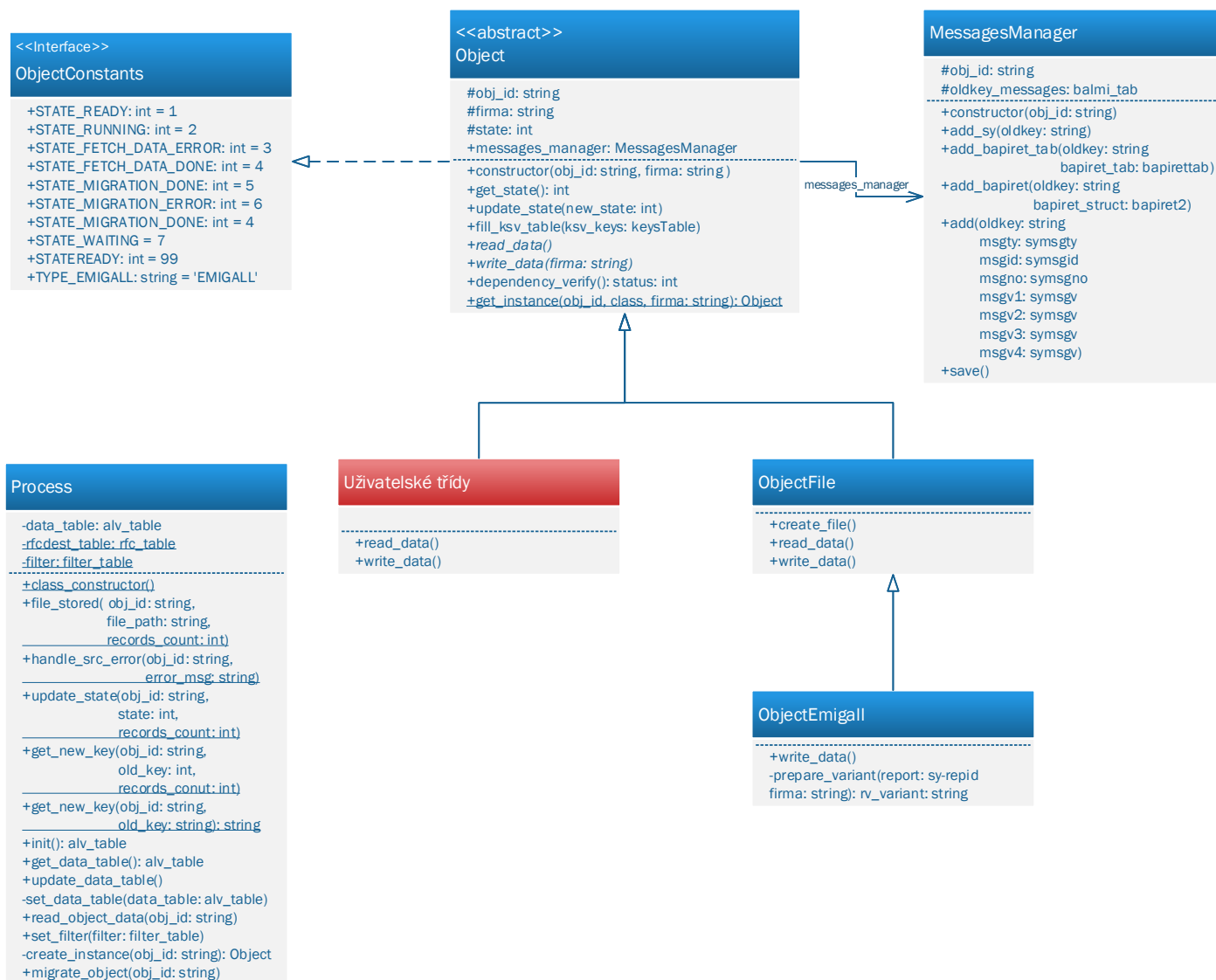
Realizace zobrazení migračního souboru je navržena tak, že bude využito možnosti odchytávání dvojkliků nad jednotlivými grafickými prvky. Zobrazení souboru je tedy navrženo na dvojklik nad načteným, případně již migrovaným objektem. Na tuto akci by měla být zobrazena obrazovka s daty migračního souboru.

Na obrázku 5.2 je vidět navržená struktura prezentovaných dat. Klíčovým atributem této struktury je pak atribut Třída. Tento atribut reprezentuje typ třídy, který odpovídá konkrétnímu migračnímu objektu.

5.1.4 Reprezentace migračních objektů

Přestože je jazyk ABAP poměrně starý a vyvinul se z jazyka COBOL (více v kapitole 6), umožňuje moderní verze jazyka ABAP objektově orientovaný přístup. Dovoluje tedy stejně jako například v jazyce Java apod. vytvářet třídy se všemi tradičními vlastnostmi, jako je dědičnost, abstrakce atd. Pro reprezentaci migračních objektů je tedy vhodné využít tento přístup, neboť bude zapotřebí rozlišit různé typy objektů, které však budou mít velice podobný základ. Na obrázku 5.3 je navržen odpovídající diagram tříd. Princip reprezentace je založen na tom, že je navržena jedna abstraktní třída, která reprezentuje obecný předpis pro všechny typy objektů. Framework obsahuje právě jednu takovou abstraktní třídu (Object) a její podtřídu ObjectFile reprezentující obecně objekt migrovaný pomocí

binárního souboru. Její vlastnosti potom dědí třída `ObjectEmigall`. Ta reprezentuje migrační objekt, pro jehož migraci lze využít metody nástroje EMIGALL. Je specializací třídy `ObjectFile`, neboť EMIGALL pracuje také s binárním souborem, ale jde o specifitější objekt. Jednotlivé instance těchto tříd bude zapotřebí řídit a spravovat. Pro tento účel je navržena separátní třída `Process`. Tato třída je také hlavním zdrojem dat pro obslužný program migrace (viz podkapitola 5.1.3).



Obrázek 5.3: Diagram tříd.

Na obrázku 5.3 jsou tedy znázorněny již zmíněné třídy. Kromě nich je v diagramu znázorněna také „Uživatelská třída“ (označena červenou barvou). Tato třída není v systému

implementována. Jde o znázornění třídy, kterou si uživatel může vytvořit z abstraktní třídy **Object**. Tato úroveň tříd slouží k implementaci speciálních tříd, které nejsou realizovány skrze binární soubory. Může se jednat například o implementaci, kde import dat bude probíhat přímým zápisem do databáze. Obecná abstraktní třída **Object** obsahuje atributy **obj_id**, **firma** a **state**. Tato třída je navržena tak, aby obsahovala veškeré obecné implementace, jako je získání stavu z databáze pro daný objekt, a naopak nastavení stavu objektu s příslušným aktualizováním záznamu v databázi. Dále obsahuje definice metod, které je potřeba u potomků implementovat, konkrétně abstraktní metody **read_data()** a **write_data()**. Ke třídě **Object** je přidružena i třída **MessagesManager**, která je s třídou **Object** propojena ve formě atributu. Tato třída je využita v případě, že je potřeba zaznamenávat chyby a události v rámci zpracování migračního objektu. V případě třídy **ObjectEmigall** zůstává tato třída nevyužita, neboť **EMIGALL** již zaznamenává chyby automaticky. Jak již bylo zmíněno, v diagramu je popsána třída **Process**. Tato třída je hlavní řídicí třída, která bude sloužit jak pro řízení migračního programu, tak ke správě migračních objektů.

Třída má navrženou sadu statických metod, které slouží pro správu zpětných volání ze zdrojového systému (podrobný popis bude popsán v kontextu kapitoly 6). Ostatní metody pak slouží právě ke správě jednotlivých objektů, nebo připravují data k prezentaci uživateli obslužného programu. V neposlední řadě diagram obsahuje rozhraní **ObjectConstants**. Již z názvu je patrné, že se jedná o rozhraní, které obsahuje konstanty. Tento princip se v jazyce ABAP v praxi často uplatňuje, kdy rozhraní slouží jako kontejner pro konstanty. Díky tomuto přístupu jsou pak jednotlivé konstanty na třídě shromážděny do jednoho přehledného místa. Další výhodou pak může být to, že toto rozhraní lze implementovat další jinou třídou, která pak bude mít shodnou sadu konstant. V tomto případě toto rozhraní reprezentuje veškeré stavy, kterých může stav migračních objektů nabývat.

5.1.5 Způsob řízení migračních objektů

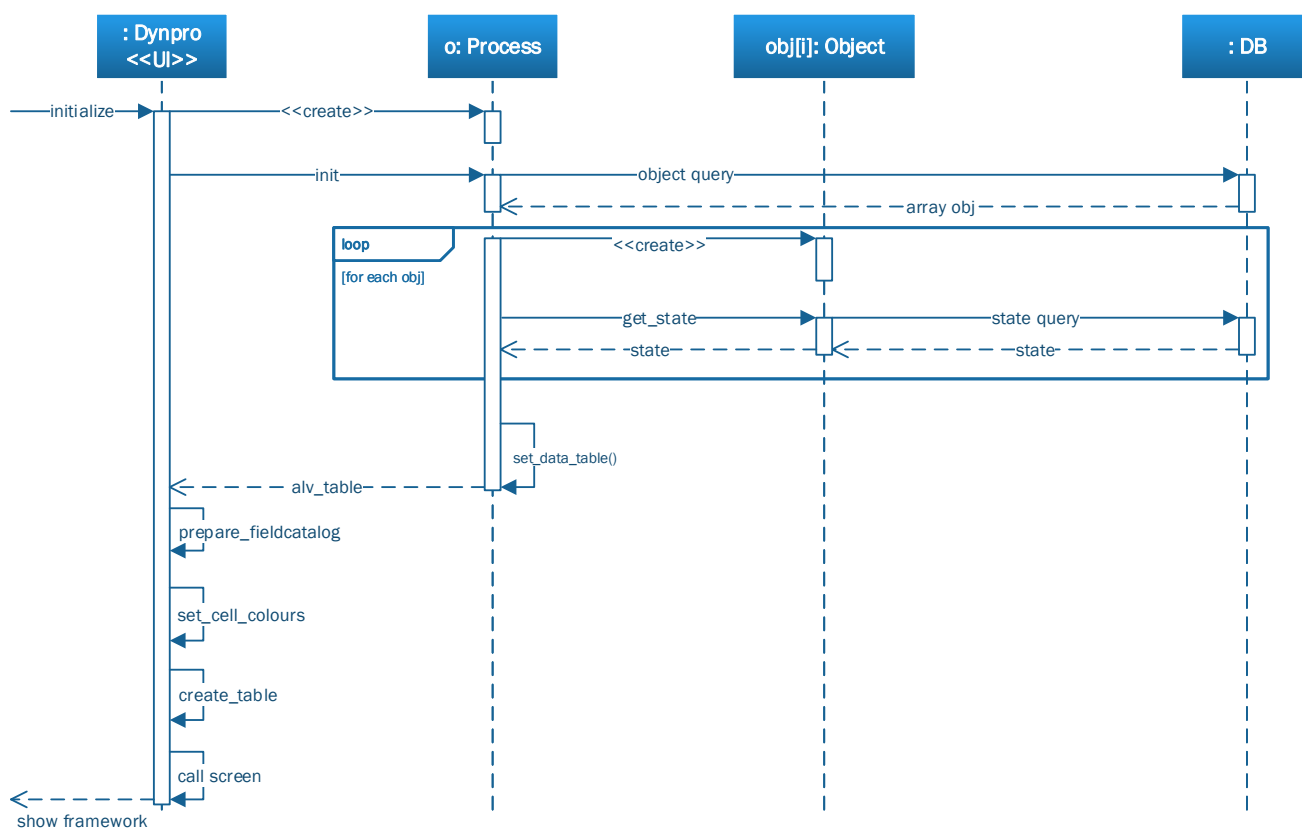
Třída **Process** má tedy za úkol řídit jednotlivé objekty. Z důvodu potřeby řídit tok migrace i v případě užití migračního frameworku více uživateli zároveň, je nutné řešit problematiku hromadného přístupu a vhodného sdílení aktuálních dat. Jako jednou z velice zajímavých variant se jeví varianta nazvaná objekt ve sdílené paměti (z anglického *Shared memory object*). Jde o realizaci sdíleného paměťového prostoru, který je vyhrazen pro uložení instance třídy, či anonymního datového objektu. Výhodou tohoto přístupu je, že paměť je sdílená pro všechny ABAP programy na daném serveru. Daná instance je tedy vytvořena pouze jednou a zůstává perzistentně uložena v této sdílené paměti, kdy k ní můžou jednotlivé programy (zde jednotliví uživatelé skrze obslužný program) přistupovat a používat její metody. Tato sdílená paměť je obohacena o vlastní zámky přístupu. Avšak bohužel z oficiální dokumentace se lze dočíst, že tento přístup je vhodný pouze pro specifické problémy. Doslova se nedoporučuje používat tuto techniku, očekáváme-li hodně paralelních čtecích a zapisujících přístupů, či dokonce plánujeme hodně zapisovat do takto sdílené instance. Při základním otestování tohoto přístupu se tato doporučení doopravdy naplnila. Další nevýhodou je, že při použití sdíleného objektu nelze mít v třídě atribut jako referenci na jinou třídu, tedy pro použití sdílené paměti nesmí být žádný atribut ukazatelem [7, 24].

Nabízí se také řešení, ve kterém by třída **Process** měla jako atribut pole (v jazyce ABAP tabulku s jedním sloupcem) referencí na jednotlivé instance reprezentující migrační objekty. V tento moment bychom ale museli stejně vyhledávat instance v seznamu a aktualizovat

jejich stavy a jejich ekvivalentní záznamy v databázi (databázové schéma viz podkapitola 5.1.6).

Nakonec je řízení instancí jednotlivých tříd reprezentujících migrační objekty navrženo tak, že bude v případě použití konkrétních metod daných tříd vytvořena „ad-hoc“ instance pouze na nezbytně nutnou dobu, kdy při změnách svého stavu změní také příslušnou databázovou tabulku reprezentující stavy tohoto objektu. Při zapisování změny stavů lze při implementaci pomocí jazyka ABAP vytvořit tzv. zámky, kterými lze řešit souběžný zápis do databáze. Tento postup je zvolen z důvodu, že instance pouze vyvolávají akce, ale přímo nečekají na odpověď a samotné provádění takovýchto akcí může trvat klidně déle než hodinu (extrakce dat ze zdrojového systému, importování dat do cílového) a tedy není nutné si uchovávat instance po celý běh programu.

Návrh komunikace takto popsaných objektů je popsán na následujícím obrázku, který znázorňuje inicializaci migračního frameworku, resp. zapnutí obslužného programu.



Obrázek 5.4: Sekvenční diagram inicializace programu.

Sekvenční diagram popisuje interakci mezi jednotlivými třídami při spuštění programu. Při spuštění frameworku dojde k vytvoření instance třídy **Process**. Tato instance pak z databáze načte předem specifikované informace o migračních objektech. V těchto informacích je obsažena i informace o tom, které třídy odpovídají těmto objektům. Na základě této informace si vytvoří instance těchto objektů. Instance migračních objektů pak do instance třídy **Process** předávají všechny další potřebné informace, stěžejní je informace o stavu objektu. Tyto informace jsou pak v třídě **Process** zpracovány do podoby tabulky, která je použitelná pro grafické uživatelské rozhraní. Grafické uživatelské rozhraní pak inicializuje

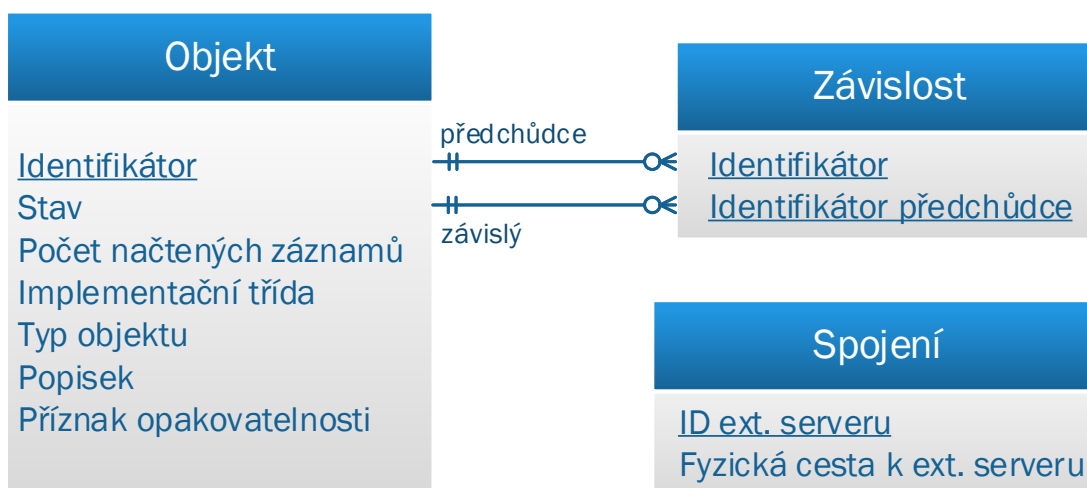
všechny grafické prvky. Po této inicializaci se uživateli zobrazí obslužná obrazovka migračního frameworku.

Framework obsahuje také třídu `Event_class`, která není obsažena ani zde v sekvenčním diagramu, ani v diagramu tříd. Tato třída využívá standardní ABAP přístup nazvaný `SET HANDLER/EVENT HANDLER` [6]. Díky tomuto přístupu můžeme ve třídě implementovat metodu, která je zavolána pouze pokud byla vyvolána událost daného typu, která byla třídě přiřazena při definici. Třída tedy bude reagovat na některé specifické uživatelské akce, jako je například dvojklik provedený nad tabulkou s informacemi o migračních objektech v obslužné obrazovce. Vyvolání a samotné odchytnutí uživatelské události je už pak řešeno automaticky na aplikační vrstvě systému SAP. V neposlední řadě je ve schématu vyznačena databáze. Detaily této databáze jsou prezentovány v následující podkapitole.

5.1.6 Databázové schéma frameworku

Jazyk ABAP byl stvořen na míru pro účely firmy SAP, a to převážně k realizaci informačních systémů a pozdějších pokročilých ERP systémů. Tyto systémy pracují především s daty, která jsou uložena v databázi. Z tohoto důvodu má jazyk ABAP vestavěnou podporu pro jazyk SQL (tedy OpenSQL přímo v syntaxi jazyka). Kromě využití této vestavěné podpory pro jazyk SQL dovoluje ABAP zadávat přímo SQL dotazy pro danou databázi, která je na databázové vrstvě nainstalována. Dokonce zakládání nových tabulek včetně pokročilých operací nad těmito tabulkami lze provádět přímo ze SAP GUI. Díky tomuto faktu jsou veškerá data k různým programům ukládána vždy do databázových tabulek. Pro potřeby frameworku je na straně cíle navržena poměrně jednoduchá databáze, která slouží k zaznamenávání jak uživatelských dat¹, tak běhových dat².

Na následujícím obrázku je znázorněn jednoduchý konceptuální model:



Obrázek 5.5: Konceptuální model (cílový systém).

V kontextu konceptuálního modelu lze říci, že potřebujeme mít možnost uchovávat informace o jednotlivých migračních objektech a k nim moci pouze definovat závislé objekty.

¹ V prostředí SAP se běžně používá pojem „customizační“ data a následně „customizační“ tabulka.

² Označovaná jako aplikační data a následně pak aplikační tabulka.

Mimo to systém potřebuje mít definovaná spojení. Tato spojení pak reprezentují možná spojení na zdrojové systémy. Spojení jsou nastavena na úrovni uživatelského nastavení systému SAP tak, jak bylo představeno v podkapitole 4.2. Entita Spojení z konceptuálního modelu pak reprezentuje překladovou tabulku pro překlad uživatelsky přívětivě zadané adresy na složitě zapsanou fyzickou adresu k externímu serveru.

V prvním odstavci této podkapitoly byly zmíněny pojmy uživatelská a běhová data. Jedná se o standardní zavedené pojmy z pohledu databáze v SAP řešení. Těmto pojům odpovídají 1 : 1 pojmy uživatelská tabulka a aplikační tabulka³. Uživatelskou tabulkou označujeme takovou tabulku, která není plněna v průběhu běhu programů, ale je naplněna uživatelsky. K tomuto plnění lze k těmto uživatelským tabulkám vygenerovat tzv. náhled údržby (z anglického maintenance view), díky kterému pak může uživatel snadno skrze SAP GUI nastavovat obsah této tabulky. Výhoda takového postupu je v tom, že lze tyto tabulky využít k nastavování určitých definičních hodnot, které může jednoduše naplnit i uživatel systému SAP, který nemá znalosti z jazyka ABAP či SQL. Oproti tomu aplikační tabulky, jak již napovídá název, slouží k ukládání dat, která vznikají při běhu programů.

Z důvodu zmíněného členění na uživatelské a aplikační tabulky je v prostředí SAP poměrně běžně vyskytující se vztah 1 : 1, který je způsoben právě tím, že do uživatelských tabulek by nemělo být zasahováno z programu, a naopak do aplikačních by neměl zasahovat uživatel. Tedy uživatel skrze SAP GUI nastaví pouze některé definiční hodnoty, ale jakékoliv další informace, které vznikají při běhu programu, jsou pak zaznamenány do aplikačních tabulek, které jsou s touto danou uživatelskou tabulkou spojeny.

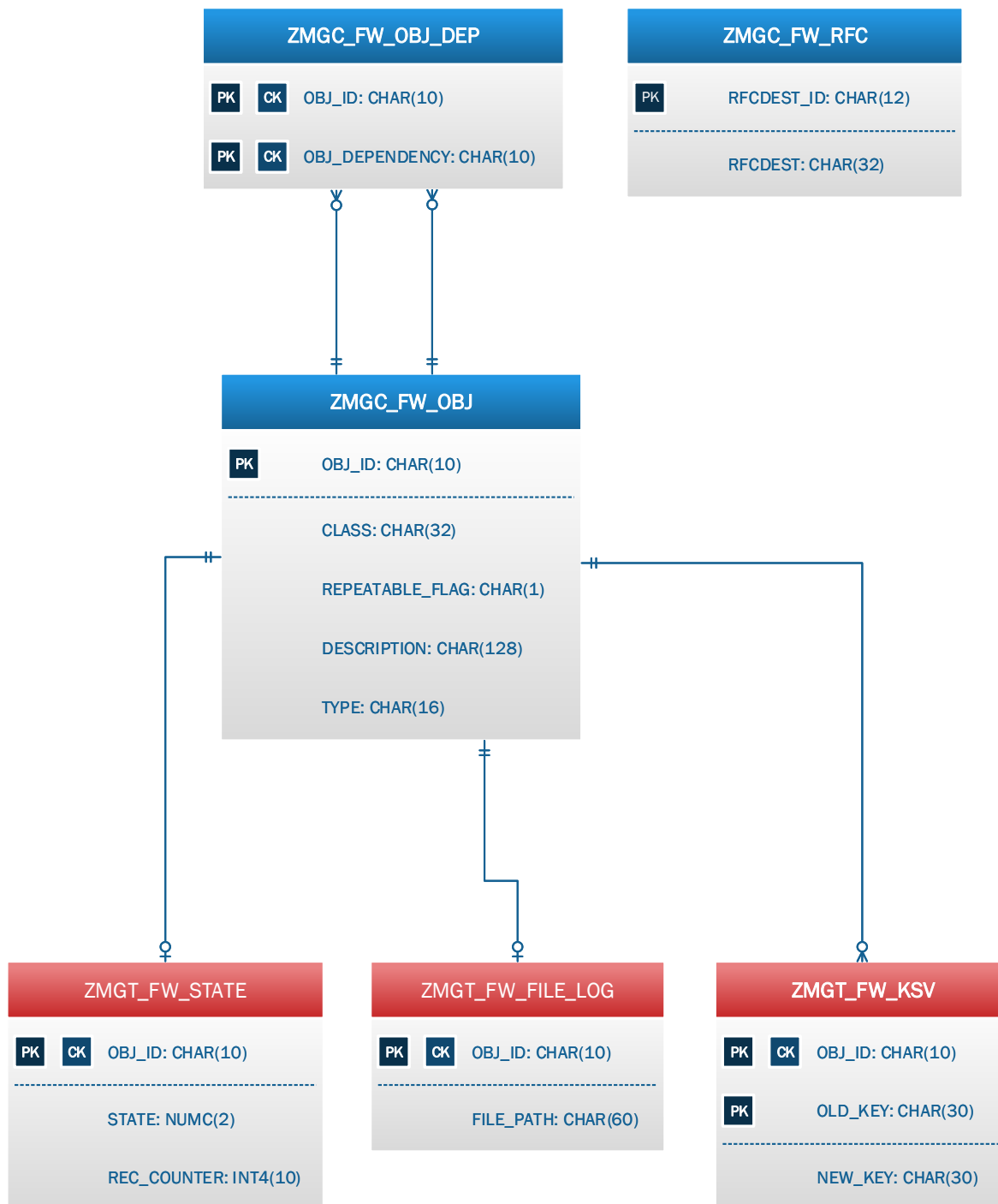
Dále je vhodné zmínit specialitu databáze v prostředí SAP. V drtivé většině tabulek je účast atributu MANDT. Jedná se o třímístný číselný identifikátor, který označuje tzv. klienta. V systémech SAP je jeden systém často rozdělen do více klientů⁴. Klientem se zde myslí jakási vrstva daného systému. Můžeme mít tedy například systém XXX a k němu je definovaný například klient 100 a klient 200. Každý klient má pak svoji vlastní sadu kmenových dat a vlastní sadu tabulek, resp. data jsou rozdělena právě pomocí hodnoty MANDT. Tato hodnota se plní do tabulek automaticky při volání databázového příkazu `insert` či `modify` nad databázovou tabulkou a programátor je tedy značně od tohoto faktu odstíněn. Přesto však má ale jisté prostředky, jak toto rozdělení na základě hodnoty MANDT obejít a načíst i data pro jiného klienta, než ve kterém je aktuálně přihlášen. Rozdělení klientů pak může mít důvod například k oddělení vývojového a testovacího systému, nebo například rozdělení systému pro rozdílné kontinenty apod. Pak se tedy nemůže stát, že by programátor při novém vývoji omylem nějakým způsobem narušil data, která jsou např. již v produkční vrstvě [11].

Na následující straně je na obrázku 5.6 znázorněno fyzické databázové schéma. Pro následující obrázek budeme předpokládat takové označení, že uživatelské tabulky jsou znázorněny modře a aplikační pak červeně. Klíčovou tabulkou zůstává stejně jako v konceptuálním modelu uživatelská tabulka reprezentující migrační objekt. Ve fyzickém modelu je však doplněna o již zmíněná specifika řešení SAP, která zapříčinila rozpad tabulky na více malých tabulek. Jedná se o aplikační tabulku stavů `ZMGT_FW_STATE`. Dále pak můžeme ke každému objektu mít uloženou cestu (aplikační tabulka `ZMGT_FW_FILE_LOG`) k binárnímu souboru. U tříd, které nepracují s binárním souborem, je však tato tabulka nevyužita. Aplikační tabulka `ZMGT_FW_KSV` slouží ke správě klíčů migračních objektů. Objekty budou totiž zpra-

³ V případě aplikační tabulky je tento název srozumitelný, ale v případě „customizační“ tabulky bude v tomto textu používán pojem uživatelská tabulka.

⁴ Bohužel pojem klient je matoucí, ale nejedná se o klienta ve smyslu zákazníka apod., ale spíše o instanci daného systému.

vidla v novém systému mapovány na rozdílný klíč. V případě objektů, které budou migro-
vány za pomoci nástroje EMIGALL je automaticky využita a plněna obdobná standardní
tabulka TEMKSV.



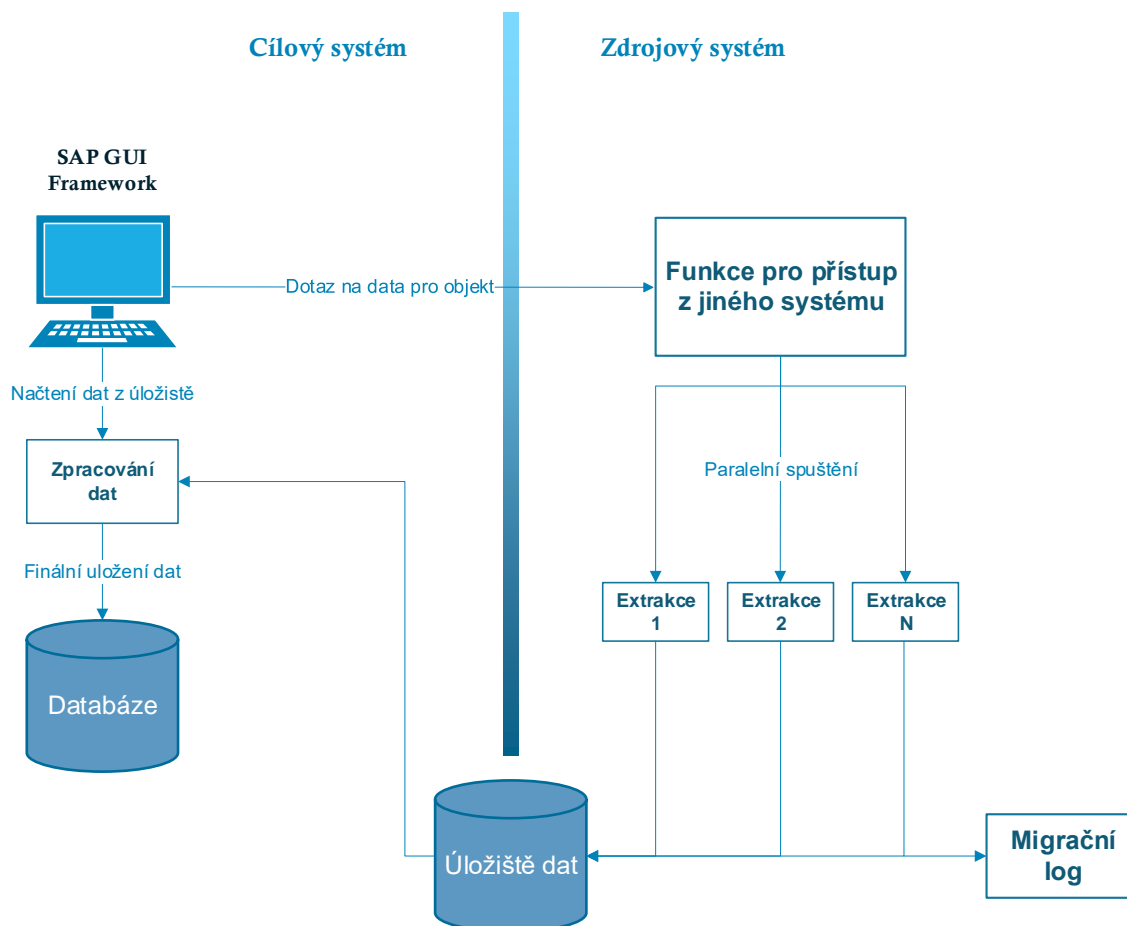
Obrázek 5.6: Fyzický model databáze (cílový systém).

5.2 Zdrojový systém

Doposud byl popsán pouze návrh frameworku na cílovém systému. Nicméně už nyní byla zmíněna například tabulka spojení, kde jsou definována spojení k jiným systémům, kterými se zpravidla myslí zdrojové systémy, tedy systémy obsahující data, která je zapotřebí migrovat na jiný systém. V případě restrukturalizace podniku popsané v kapitole 4 může těmito systémy být jak SAP/ISU, tak například i SAP/CRM. Tyto systémy slouží tedy převážně k načtení zdrojových dat. Při načítání se může stát, že z důvodu zákaznických požadavků dojde k omezení některých dat. Například může zákazník specifikovat, že nechce migrovat historické měřicí přístroje, které však jsou s partnerem (zákazníkem společnosti) spojeny, nebo například nechce migrovat smluvní účty, které nesplňují určité parametry. Přesto však je nutné takto nenačtená data zaznamenat včetně jejich důvodu k vyřazení z migrace. Hlavním prvkem na zdrojovém systému jsou však extrakční funkce, jejichž návrh je představen v následujícím textu.

5.2.1 Komunikace mezi systémy

Na následujícím obrázku lze vidět návrh technického řešení extrakce zdrojových dat:



Obrázek 5.7: Technické schéma systému.

Systém SAP umožňuje poměrně jednoduchý způsob spojení mezi systémy. Aby bylo možné popsat tento způsob, je nejprve nutné vysvětlit pojem funkční modul. V systému SAP lze do objektového repositáře, jakéhosi globálního systémového repositáře vývojových objektů, založit kromě programu i tzv. funkční modul. Funkční modul je vlastně funkce taková, jakou ji známe například z jazyka C. Stejně tak jako tyto funkce, může mít funkční modul definované vstupní a výstupní parametry. Velký rozdíl je však v tom, že má tento funkční modul definici i implementaci v již zmíněném globálním repositáři a je možné jej volat z kterékoliv další implementace (např. dalšího funkčního modulu či programu). Při zakládání tohoto funkčního modulu lze explicitně určit, že tento funkční modul je možné volat pomocí tzv. RFC [24], což je zkratka z anglického **remote function call**, tedy česky přeložitelné jako volání vzdálené funkce. Tento mód předpokládá, že tato funkce bude mimo jiné volána z jiných SAP serverů, než na kterých je implementována. Takto označená funkce vyžaduje, aby předávané parametry byly vždy předávané hodnotou a nepodporuje předání reference na data. Pokud je tedy definované spojení mezi SAP serverem a funkcí je označena jako RFC, lze ji s použitím klíčového slova **DESTINATION** (česky cíl) a cestou k tomuto serveru zavolat. Zde se dostáváme k významu tabulky spojení, tedy cest, z minulé podkapitoly.

Právě této vlastnosti je využito při návrhu schématu, jak je možné vidět na obrázku 5.7. Funkce pro přístup z jiného systému je navržena právě jako takováto RFC funkce. Tato architektura předpokládá asynchronní volání této funkce, a to s minimem dat, kdy funkce poté vždy pouze spustí paralelní načtení dat konkrétních objektů.

Jednotlivé extrakční funkce tak, jak jsou znázorněny na schématu, nejsou jako takové součástí migračního frameworku. Při využití frameworku na jednotlivých migračních projektech se předpokládá vždy nová sada extrakčních funkcí. Neboť se předpokládá velké množství extrakčních funkcí, které navíc budou vyvíjeny mnoha různými vývojáři a někdy i mnoha různými týmy, je v rámci návrhu migračního frameworku nutné zaměřit se na obecné procesy při extrakci dat, které je nutné provádět vždy.

5.2.2 Extrakční proces

Extrakčním procesem se rozumí proces vytvoření takového formátu dat ze zdrojového systému, aby jej bylo možné odeslat do cílového systému, kde budou takto načtená data zpracována a uložena do databáze. Nejčastějším formátem dat v prostředí SAP/ISU, konkrétně pak jeho odvětvového řešení pro společnosti zabývající se výrobou a obchodováním s elektřinou, které je stěžejní pro tuto práci, je formát binárního souboru se strukturou, kterou očekává EMIGALL.

EMIGALL formát binárního souboru

Aby bylo možné data importovat, musí odpovídat přesně formátu tak, jak jej EMIGALL očekává. Po analyzování tohoto souboru (soubor s prázdnými daty lze pro libovolný objekt vytvořit přímo v nástroji EMIGALL) bylo zjištěno, že soubor má strukturu definovanou tak, jak je vidět na obrázku 5.8.

Na tomto obrázku je možné vidět strukturu binárního souboru pro objekt **CONNBJ**, který reprezentuje přípojný objekt. Soubor obsahuje záznamy pro přípojný objekt, konkrétně pak tři fiktivní přípojně objekty ve městě Brně. Nejpodstatnější je fakt, že jednotlivé záznamy obsahují vždy data pro jednotlivé záznamy objektu pro daný klíč. Typ informace pro jednotlivé záznamy je pak určen pomocí **DTyp**. Tyto typy se definují přímo v nástroji EMIGALL. Proces nastavování tohoto nástroje je však poměrně složitý a přesahuje rozsah této práce. Pro potřeby této diplomové práce postačí pouze informace, že pro jednotlivé

objekty jsou tyto dílčí typy nastaveny pro každý objekt. Typy nemusí být však vždy vyplněny. Například na obrázku je názorné, že záznamy s klíčem 76 a 77 nemají žádná telefonní čísla, tedy typ CO_COM. Vždy je však nakonec přítomen ještě typ „&ENDE“. Tento typ je nezbytný pro funkci zpracování souborů, kdy je vyplněn pouze klíč a typ. Dále u každého jednotlivého záznamu je nutné před exportem vypočítat délku bytového řetězce, resp. tedy délku celého záznamu. Z obrázku sice není tato informace patrná, ale každý soubor musí obsahovat také hlavičku „&INFO“. V této hlavičce lze vyplnit údaje o uživateli, který soubor vytvořil apod.

Firma: Z_Test		Název: HANA KT	
MigObj: CONNOBJ		Date: 11.01.2018 Time: 09:58:22	
Délk	Klíč dosav.systému	DTyp	Datová část
0055	75	CO_EHA	CO Test VO 02 PL_1
00A4	75	CO_ADR	Brno 623 00 Šárka
0076	75	CO_COM	777666555
0076	75	CO_COM	608666555
0026	75	&ENDE	
0055	76	CO_EHA	CO Test VO 03 PL_1
00A4	76	CO_ADR	Brno 623 00 Bašného
0026	76	&ENDE	
0055	77	CO_EHA	CO Test VO 04 PL_1
00A4	77	CO_ADR	Brno 602 00 Jílová
0026	77	&ENDE	
FFFF			

Obrázek 5.8: Binární soubor pro EMIGALL.

Extrakční funkce

Na základě této analýzy je navržen postup, kterým je umožněno z předem definované interní ABAP struktury vytvořit právě takovýto binární soubor. Tato funkce by měla být následně volána v každé extrakční funkci, která je určena pro typ migrace EMIGALL. Tento nástroj při definování struktury migračního objektu také generuje struktury, které odpovídají jednomu záznamu ze souboru. Tyto struktury je nutné manuálně přenést do zdrojového systému ke každé odpovídající extrakční funkci. Takto naplněné struktury budou předány funkci, která bude schopna zajistit správné vytvoření binárního souboru. Tento formát souboru lze využít i pro vlastní implementaci přenášení dat pomocí souboru. Tato část je tedy obecná pro celou třídu objektů `ObjectFile` (podkapitola 5.1.4), tedy i její případné potomky.

Nástroj EMIGALL je dostupný pouze v našem konkrétním odvětvovém rozšíření, tedy SAP/ISU. V prostředí SAP/CRM je nutné se obejít bez tohoto nástroje. Pokud není využito na straně cíle třídy `ObjectFile`, je možné implementovat abstraktní třídu `Object`, která bude obsahovat vlastní metody, které zajistí extrakci a následné uložení dat. Na straně zdroje je pak opět potřebné vytvořit extrakční funkční modul, který bude spuštěn z přístupového bodu (RFC funkční modul). Může se jednat například o funkční modul, který zavolá přímo nějaký definovaný přístupový bod na straně cíle, který rovnou zapisuje do databáze data.

Jednotlivé vyřazené záznamy objektů

Již bylo zmíněno, že při extrakci dat může dojít k vyřazení některých záznamů. Zákazník může specifikovat různá omezení, kontrolu kvality dat a další různé závislosti. Pokud není některá z těchto podmínek splněna, je nutné zaznamenat, které záznamy objektu byly z migrace vyřazené, a to včetně důvodu vyřazení. Tyto záznamy pro jednotlivé klíče budou zaznamenány do tzv. migračního logu, který bude obsahovat jak migrované, tak i právě popsané vyřazené záznamy.

Logování načtených zdrojových objektů

Logování načtených zdrojových objektů

ID obj.	Key	Icon	Text hlášení	Typ ...	Komodita	Počet migrovaných	Počet vyřazených
PARTN...	1	❌	OP označen k výmazu	FO	GAS		1
PARTN...	12	✅		PO	ELE	1	
PARTN...	13	✅		PO	ELE	1	
PARTN...	14	❌	OP označen k výmazu	FOP	ELE		1
PARTN...	145	✅		FOP	ELEGAS	1	
PARTN...	15	✅		FOP	ELE	1	
PARTN...	16	✅		FOP	ELEGAS	1	
PARTN...	17	✅		FOP	ELEGAS	1	
PARTN...	18	✅		FOP	ELEGAS	1	
PARTN...	2	✅		FO	GAS	1	
PARTN...	222	✅		PO	ELE	1	
PARTN...	3	✅		FO	GAS	1	
PARTN...	3659	✅		PO	ELE	1	
PARTN...	4	✅		FO	GAS	1	
PARTN...	4557	✅		PO	ELE	1	
PARTN...	5	✅		PO	ELE	1	
PARTN...	56	✅		FO	GAS	1	
PARTN...	599	✅		PO	ELE	1	
PARTN...	788	✅		PO	ELE	1	
PARTN...	8	✅		FO	GAS	1	
PARTN...	9	✅		FO	ELE	1	
PARTN...	988987	❌	OP nemá smlouvu	PO	ELE		1
PARTN...	989	✅		FO	GAS	1	
						20,0	3,0

Obrázek 5.9: Detailní migrační log.

Obrázek 5.9 představuje návrh migračního logu, resp. jeho prezentaci uživateli. Pomocí tohoto programu, který bude vyvinut na straně zdroje, si bude moci uživatel, zde zpravidla zákazník či migrační technik, prohlédnout úspěšnost extrakce dat již na zdroji. Kromě toho uvidí, jaké konkrétní objekty selhaly a přesný důvod. Ve velkém množství případů je ale mnohem více vypovídající informace o tom, kolik objektů se nepodařilo extrahovat z jednoho konkrétního důvodu. Z tohoto důvodu je migrační log rozdělen do dvou módů zobrazení. Tzv. detailní log byl představen právě na obrázku 5.9.

Oproti tomu právě popsaný tzv. sumarizační log lze vidět na následujícím obrázku:

ID obj.	Icon	Text hlášení	Typ ...	Komodita	Attr. 3	Attr. 4	Počet migrovaných	Počet vyřazených
PARTNER	❌	OP označen k výmazu	FO	GAS				1
PARTNER	✅		PO	ELE			9	
PARTNER	❌	OP označen k výmazu	FO	ELE				2
PARTNER	✅		FOP	ELEGAS			4	
PARTNER	❌	OP nemá smlouvu	FOP	ELE				1
PARTNER	✅		FO	GAS			6	
PARTNER	✅		FO	ELE			1	

Obrázek 5.10: Návrh sumarizačního logu.

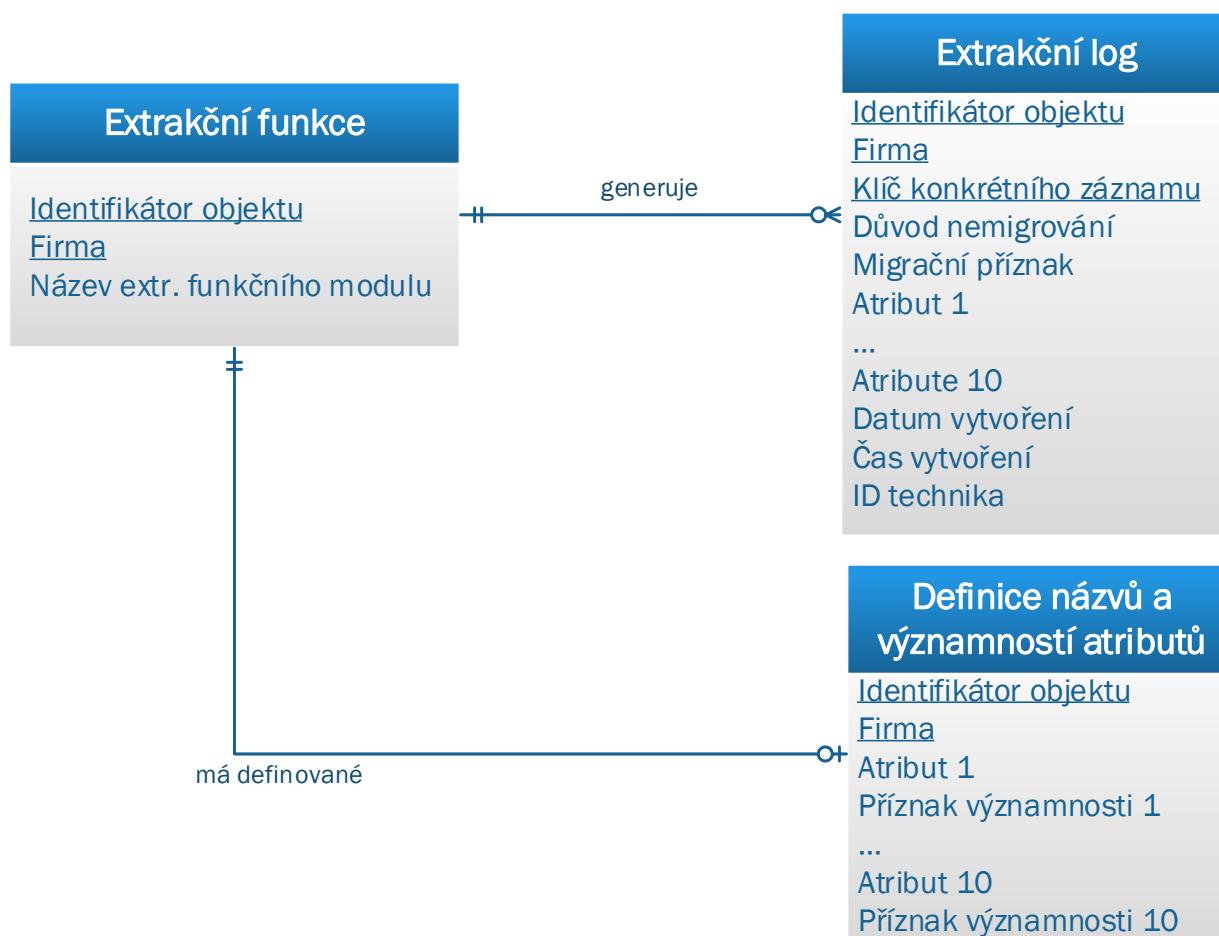
Migrační log má tedy klíčové informace, jako je ID objektu, ikona značící zdali byl objekt vyřazen nebo migrován, text hlášení a číselné statistiky. V případě, že se bude uživatel dotazovat pouze na záznamy jednoho konkrétního migračního objektu, je požadavek na to, aby ta část logu mezi textem hlášení a početními statistikami byla tvořena dynamicky a mohla být uživatelem specifikována bez nutnosti zasahování do samotné implementace v jazyce ABAP. Kromě toho v případě sumarizačního logu může uživatel nadefinovat tzv. významné atributy. Těmito atributy se rozumí atributy, které budou použity pro tvoření konkrétních sumovaných skupin, tedy aby uživatel mohl například stanovit, že významným atributem na příkladu z obrázku 5.10 je atribut Typ a Komodita. Při následném tvoření skupin je tato množina atributů poté brána společně s textem chybové hlášky jako kritérium seskupení. Tento jev je vidět na dvou chybových hláškách; v případě, že bychom měli definovaný pouze Typ jako významný atribut, by došlo ke sloučení prvního a třetího záznamu podle textu hlášení a atributu Typ. Všechny nevýznamné atributy jsou pak v tomto případě ignorovány, jak je nastíněno demonstrací pomocí atributů Attr. 3 a Attr. 4.

Na obrázku 5.10 můžeme vidět tedy dva pojmenované atributy (Typ a Komodita) a dva nezadefinované (Attr. 3 a Attr. 4). Požadavkem na tyto atributy bylo, aby jich mohlo být až 10 a jejich nastavení je představeno v podkapitole zabývající se databázovým modelem na straně zdroje.

5.2.3 Pomocná databáze na straně zdroje

Stejně jako na straně cílového systému je pro potřeby uchovávání informací, které byly popsány v minulých kapitolách (seznam extrakčních funkcí pro objekty, běhová data pro migrační statistiku, definiční data pro statistiku), navržena malá databáze.

Konceptuální model tak, jak je představen na obrázku 5.11, je opět poměrně jednoduchý. Obsahuje entitu extrakční funkce, která je navržena jako vzorník pro extrakční funkce. Funkce pro přístup z jiného systému, jak byla prezentována na obrázku 5.7, bude přistupovat právě k funkcím, které budou specifikovány v této entitě. Extrakční log a Definice názvů a významností atributů pak navazuje na předchozí podkapitolu popisující způsob

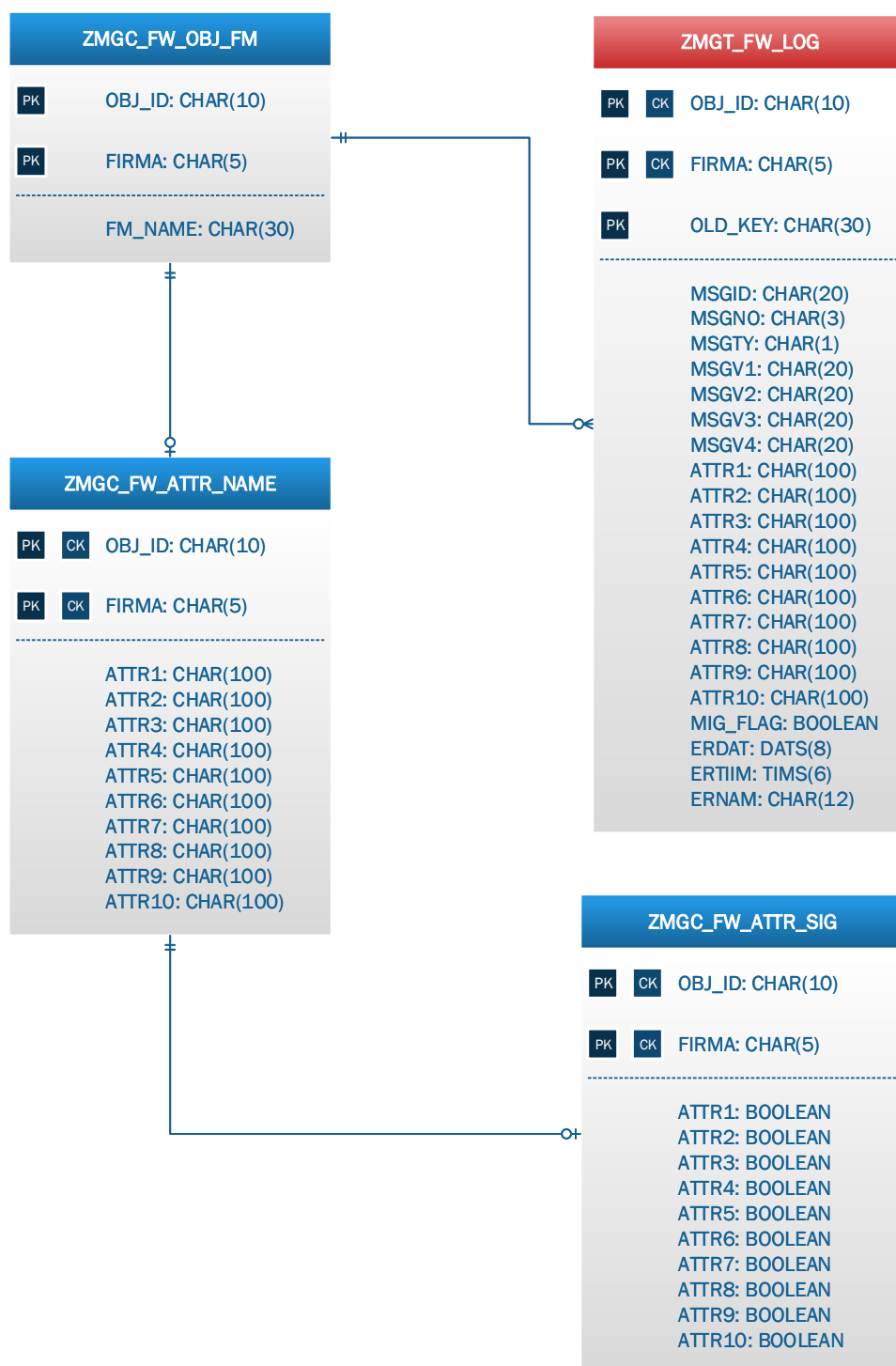


Obrázek 5.11: Konceptuální model (zdrojový systém).

zaznamenávání vyřazených migračních objektů, kdy definiční data pro uživatelskou úpravu migračního logu jsou reprezentována pomocí Definice názvů a významností atributů.

Na obrázku 5.12 je možné si všimnout, že všechny tabulky kromě červené s názvem ZMGT_FW_SRCMSG, jsou tedy tabulky uživatelské. Řešení SAP pak výrazně ovlivňuje důvod realizace dvou tabulek ZMGC_FW_ATTR a ZMGC_FW_O_SELECT, místo vytvoření jedné sloučené tabulky. Jelikož tyto uživatelské tabulky jsou nastavovány skrze standardní transakci (SM30 – pohled údržby) a pro sloučenou tabulku by tato obrazovka byla již značně nepřehledná, byla tabulka rozdělena do dvou již zmíněných, kde každá má svůj vlastní pohled údržby. Význam těchto tabulek je v nastavení atributů pro migrační log, kde tabulka ZMGC_FW_ATTR slouží k nastavení popisků konkrétního migračního objektu, zatímco tabulka ZMGC_FW_O_SELECT je doplňující tabulka, kde je uživateli umožněno zadat, které atributy jsou interpretovány jako významné a jsou na základě nich následně sdružovány jednotlivé záznamy pro objekty. Červená aplikační tabulka ZMGT_FW_SRCMSG pak slouží jako úložiště pro konkrétní data pro migrační log tak, jak byl představen na obrázcích 5.9 a 5.10.

Stejně tak, jako při popisu fyzického databázového modelu v kapitole popisující návrh na cílovém systému, je i zde vycházeno ze standardů systému SAP, tedy aplikační a uživatelské tabulky pro rozdílné typy dat, kde aplikační mají červenou barvu a uživatelské tabulky mají barvu modrou.



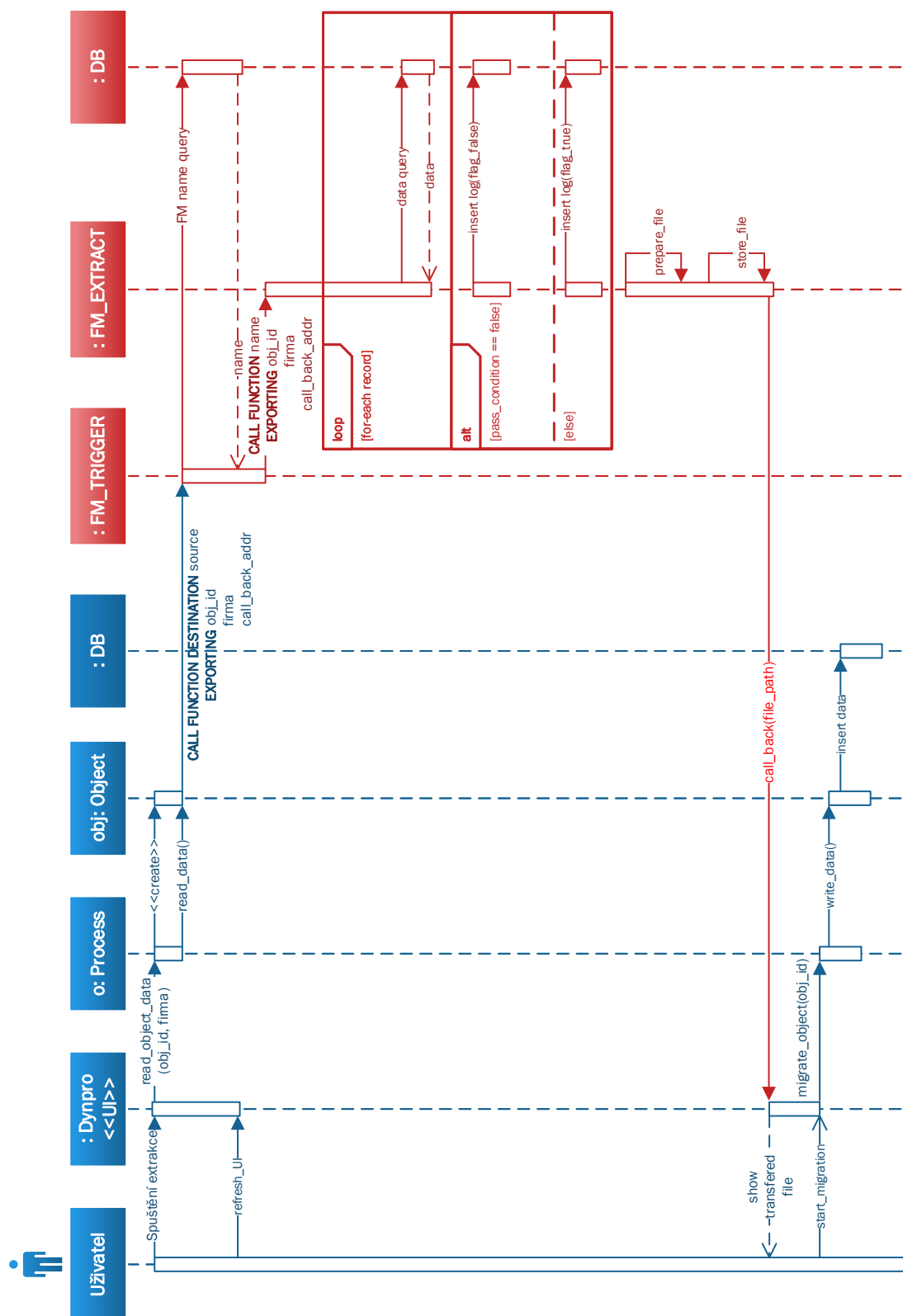
Obrázek 5.12: Fyzický model (zdrojový systém).

5.3 Návrh kompletního postupu migrace

V předchozích podkapitolách 5.1 a 5.2 byl popsán návrh implementace jak pro systém cílový, tak pro zdrojový. Na straně cílového systému byl představen návrh objektově orientovaného řešení obsluhy migrace, zatímco na zdrojovém systému byl navržen proces spuštění extrakce migračních dat bez nutnosti přihlášení k tomuto systému (vyjma procesu kontroly dat skrze extrakční log). Schéma kompletního postupu extrakce dat až po vyvolání založení dat do cílového systému je vidět na následujícím sekvenčním diagramu, kde modře je symbolizován cílový systém a červeně pak zdrojový.

Obrázek 5.13 na další straně tedy představuje ukázkou extrakce jednoho migračního objektu pomocí jedné migrační funkce. Na sekvenčním diagramu oproti inicializačnímu diagramu (5.4) můžeme nyní vidět i objekty ze zdrojového systému. `FM_TRIGGER` představuje funkční modul, který slouží pro všechny objekty jako přístupový bod z cílového systému. Dále pak `FM_EXTRACT`, který reprezentuje jednotlivé extrakční funkce na cílovém systému.

Předpokládejme, že uživatel, resp. migrační technik vybral jeden z objektů a spouští jeho migraci. Tuto akci provádí skrze standardní Dynpro obrazovku, která požadavek zpracovává. Tato obrazovka již má po inicializaci vytvořenou instanci třídy `Process`. Tato instance pak vytvoří potřebnou instanci třídy migračního objektu, která už v sobě obsahuje implementaci, ve které se volá přístupový bod na zdroji. Na zdrojovém systému dochází k načtení jména správného funkčního modulu určeného pro extrakci dat. Po spuštění tohoto modulu dochází k načtení dat migračního objektu tak, jak byla implementována programátorem používajícím tento framework. Programátor zde má možnost zaznamenávat (resp. měl by tuto činnost provádět) do migračního logu, které objekty byly úspěšně načteny, či naopak ty, které z migrace byly vyřazeny. Nakonec jsou pak data, která byla vybrána do migrace, uložena do struktury platného souboru tak, jak byl definován v podkapitole 5.2.2, nebo případně do jinak definovaného formátu. V ideálním případě dochází k uložení tohoto souboru do sdíleného úložiště. Následně dostává cílový systém zprávu o tom, kde se soubor nachází. Uživatel poté může spustit finální uložení dat do nového systému. Tato činnost probíhá opět skrze třídu `Process` a třídy reprezentující jednotlivý migrační objekt.



Obrázek 5.13: Sekvenční diagram extrakce a importace dat pomocí souboru.

Kapitola 6

Implementace frameworku

V této kapitole je představen koncept vývoje v jazyce ABAP od základního přehledu vývojových nástrojů, přes zakládání vývojových objektů, až po možné způsoby testování. Tyto souvislosti jsou uvedeny do kontextu vývoje migračního frameworku, kde je kladen důraz převážně na specifika lišící se od konvenčních programovacích jazyků.

6.1 Programovací jazyk ABAP

Jazyk ABAP je jediným programovacím jazykem použitým pro vývoj migračního frameworku (pomineme-li nativně integrovaný jazyk SQL). Jazyk ABAP byl vyvinut firmou SAP pro programování podnikových aplikací v SAP prostředí. Původně byl ABAP pouze procedurálním jazykem, ale postupem času se vyvinul v jazyk plně podporující objektově orientovaný přístup a jedná se o tzv. jazyk čtvrté generace (zkratka 4GL), který je specificky zaměřen na zpracovávání dat v podnikových aplikacích. Tento fakt potvrzují jeho vlastnosti:

- databázový přístup je integrovaný přímo do jazyka jako Open SQL,
- optimalizace výkonu přístupu k databázi integrovaná do ABAP běhového prostředí pomocí tzv. SAP buffering¹ [15],
- podpora datového typu „interní tabulka“, který slouží k dynamickému uložení a zpracování velkého objemu dat z databáze,
- koncept OLTP (Online Transaction Processing) zabudovaný přímo do běhového prostředí jazyka,
- jazyk má integrované rozhraní pro přístup z/do jiných programovacích prostředí nazvané volání vzdálené funkce (z anglického Remote function Call),
- jazyk má také integrované rozhraní pro XML [24].

Pro nové implementace, nebo i redesign starých aplikací doporučuje SAP používat všude tam, kde je to možné, pouze objektově orientovaný přístup označovaný jako ABAP Objects [14].

¹ Je schopen načíst data z databázových tabulek a ukládat je do tabulkové vyrovnávací paměti ve sdílené paměti na aktuálním aplikačním serveru.

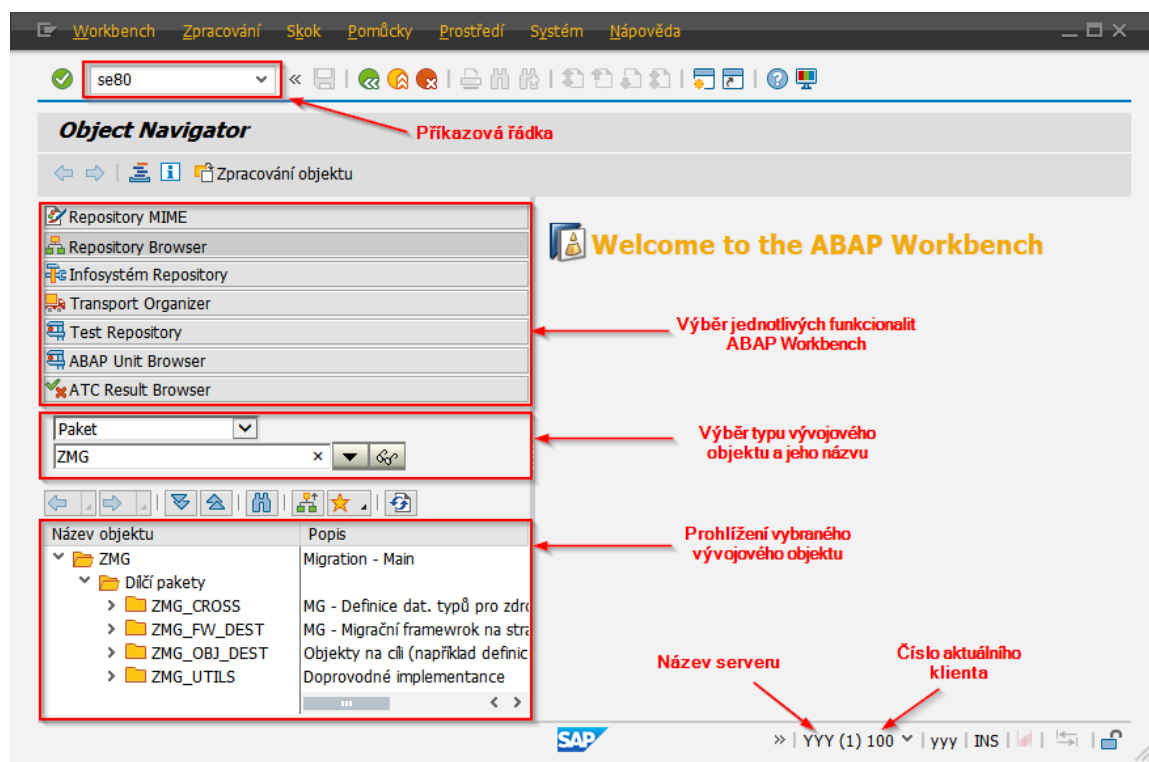
6.2 Pracovní nástroje a vývoj

Pro vývoj v jazyce ABAP se nabízí dvě následující možnosti, jak lze realizovat, resp. implementovat nějaké softwarové řešení. První možnost v podobě ABAP Workbench² [14] byla v minulosti také možností jedinou. V posledních letech se pak ale stále častěji můžeme setkávat s využitím nástroje Eclipse IDE. Toto vývojové prostředí již téměř zcela dohnalo svoji plnohodnotností tradiční Workbench.

6.2.1 SAP GUI a ABAP Workbench

Několikrát již byl zmíněn pojem SAP GUI. Pomocí SAP GUI přistupují jednotliví uživatelé k systému SAP. Tento nástroj však slouží také k samotnému vývoji v jazyce ABAP, nejenom k přístupu k již hotovým řešením. Firma SAP zde připravila standardní transakci SE80, pod kterou se skrývá ABAP Workbench. Tento pracovní nástroj lze vidět na obrázku 6.1. Slouží jako plnohodnotné vývojářské prostředí, kde lze víceméně provést vše od založení projektu, přes implementaci, až k finálnímu testování. Transakci SE80 spustíme jednoduše zadáním textu SE80 do příkazové řádky SAP GUI.

Na následujícím obrázku lze tedy vidět typické vývojové prostředí pro jazyk ABAP. Horní panel obsahující příkazovou řádku je obsažen ve veškerých, tedy snad kromě modálních, oknech. Skrze tuto příkazovou řádku je uživatel schopen pouštět různé transakce, tedy nakonec například i migrační framework.



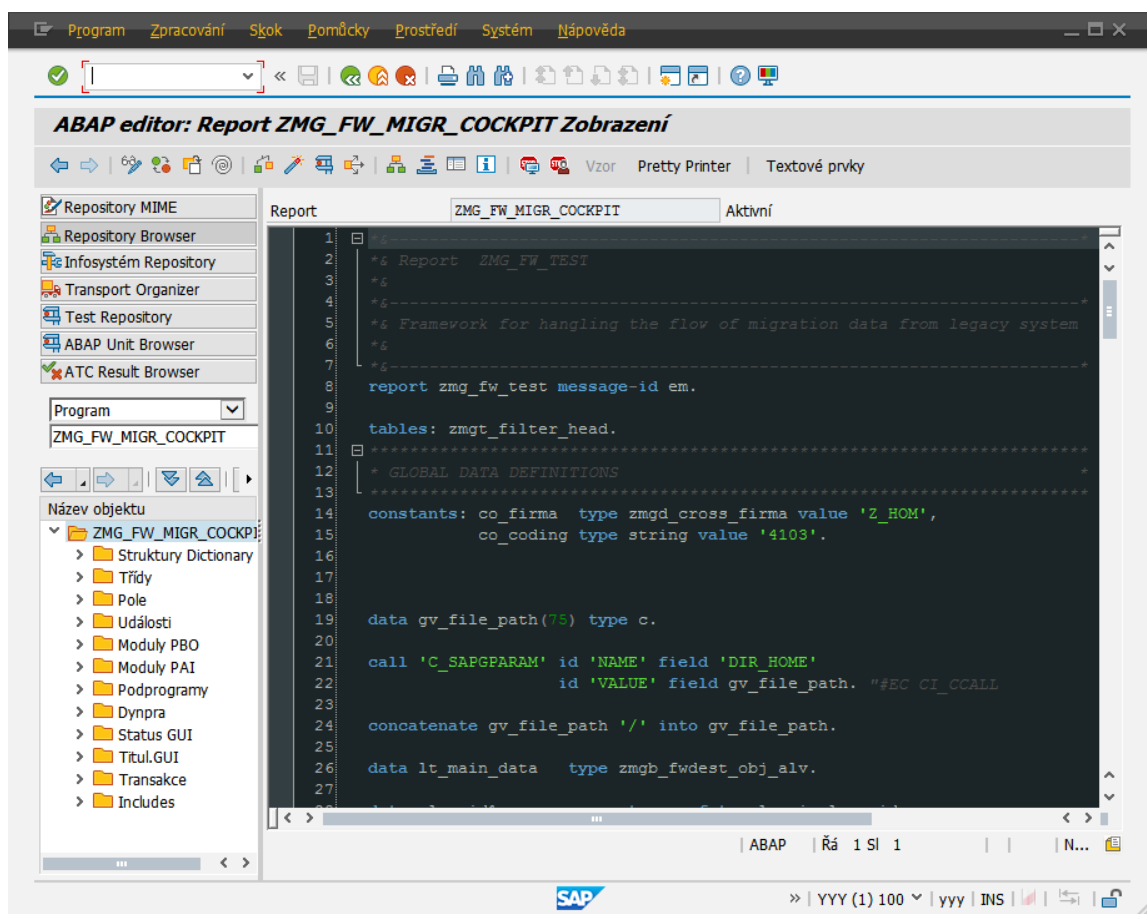
Obrázek 6.1: ABAP Workbench.

² Pojem ABAP Workbench je hojně používán i v českém prostředí, ale lze jej přeložit jako ABAP Pracovní plocha, přesto v dalším výkladu bude používán standardní pojem ABAP Workbench.

Obdobné je to i se spodním panelem, který obsahuje informace o aktuální relaci SAP GUI, tedy hlavně to, na jakém serveru se nacházíme. Hned vedle této informace je klient, a je to přesně ten stejný klient, který byl představen v podkapitole 5.1.6 popisující schéma databáze. Samotná transakce SE80 je pak vše mezitím, kde máme následující prvky:

- výběr jednotlivých funkcionalit ABAP Workbench,
- výběr typu vývojového objektu a jeho názvu,
- prohlížení vybraného vývojového objektu.

Například můžeme vybrat vývojový objekt Program a zadat název programu s implementací obslužného programu pro migrační framework. Na následujícím obrázku je vidět takto zobrazený program.



Obrázek 6.2: ABAP Workbench – vývojový objekt Program.

Tímto způsobem můžeme vyhledávat veškeré vývojové objekty, které jsou založené v tzv. repozitáři vývojových objektů, například programy, funkční moduly, pakety atd. Všechny takto vytvořené objekty jsou tedy globální a jsou přístupné i jiným vývojářům. Oproti tomu lze vytvářet tzv. lokální objekty [19].

Na pojem lokální vývoj lze narazit ve dvou rovinách. První je ta, že pojmem lokální tabulka, či například lokální třída označujeme takový vývojový prvek, který je definován pouze lokálně v programu nebo třeba ve funkčním modulu a lze ho používat pouze v tomto

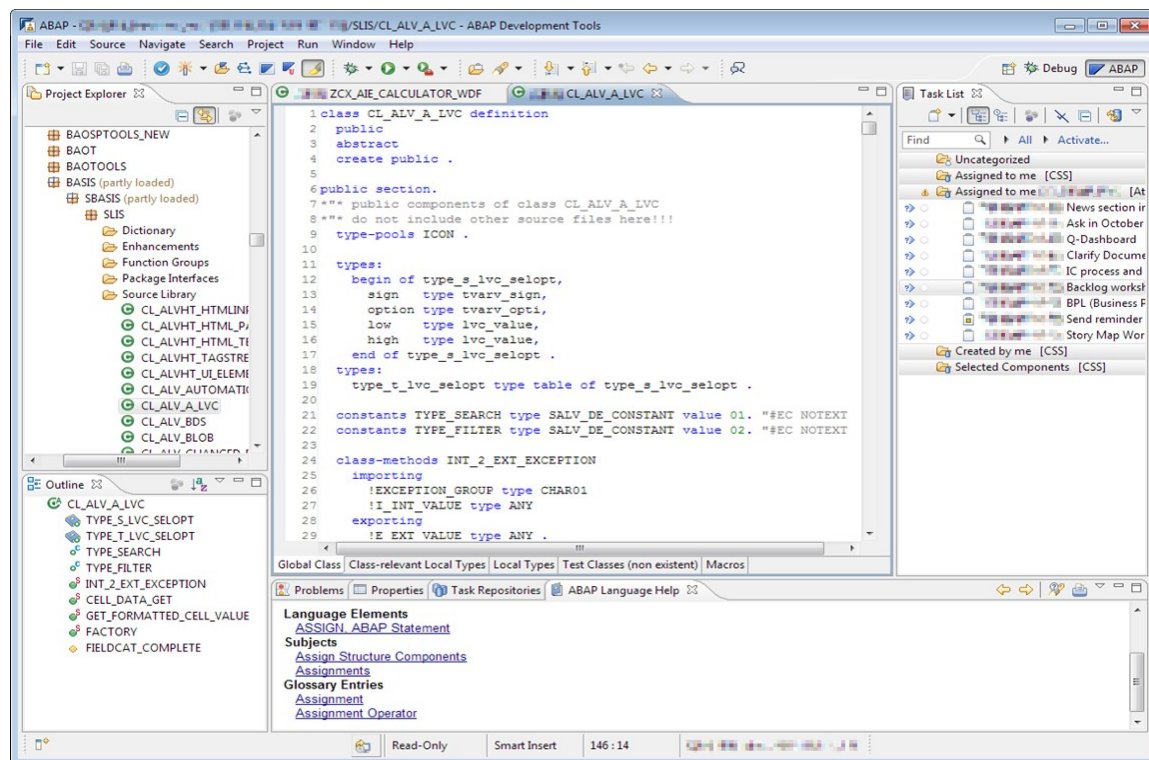
kontextu (analogicky s konvenčními programovacími jazyky). Kromě toho se lze setkat s pojmem lokální vývojový objekt, který není přiřazen k žádnému paketu a je tedy volně v systému. Přesto je tento objekt dostupný ze všech implementací i jiným vývojářům. Lokální vývojový objekt má svá specifika při transportování vývoje na jiný systém, ale toto téma přesahuje rozsah této práce.

Mezi nejdůležitější vývojové objekty v kontextu této práce patří [14]:

- paket,
- program,
- skupina funkcí (a v ní pak definované funkční moduly),
- třída,
- interface.

6.2.2 Eclipse IDE s ABAP vývojovým nástrojem

Pouze pro doplnění je zde zmíněno, že velice oblíbený nástroj Eclipse IDE [12] lze použít i pro vývoj aplikací v jazyce ABAP. Takto nastavené vývojové prostředí je vyobrazeno na následujícím obrázku:



Obrázek 6.3: Eclipse IDE s podporou ABAP [13].

6.3 Realizace navržené architektury v jazyce ABAP

Framework jako takový lze považovat za síťové řešení, ve kterém je nutné přenést data z jednoho SAP systému do druhého. Zároveň musí umožňovat zobrazení statistik extrakčních funkcí a schopnost určit následnou úspěšnost samotného založení dat v novém systému. Pro síťovou komunikaci byl zvolen přístup vzdáleně volatelných funkcí, které jsou označovány jako RFC (z anglického Remote Function Call). Z tohoto přístupu budou také vyplývat jistá technologická rozhodnutí, která jsou popsána v následujících podkapitolách.

6.3.1 RFC funkce

Framework obsahuje jednu takovouto funkci na zdrojovém systému, která slouží k vyvolání jednotlivých extrakčních funkcí. Touto funkcí je Z_MG_OBJ_DOWNLOAD_TRIGGER. Aby bylo možné volat funkci, resp. funkční modul vzdáleně (ale také například na pozadí), je nutné takto modul nastavit tak, jak je vidět na následujícím obrázku.

The screenshot shows the configuration interface for the RFC function **Z_MG_OBJ_DOWNLOAD_TRIGGER**. The interface is divided into several sections:

- Funkční modul:** Z_MG_OBJ_DOWNLOAD_TRIGGER, Aktivní
- Navigation tabs:** Vlastnosti (selected), Import, Export, Changing, Tabulky, Výjimky, Zdrojový text
- Klasifikace:**
 - Skupina funkcí: ZMG_FWSRC_OBJ, Obecné metody pro řízení migrace
 - Krátký text: Spouští načítání dat ve formě background tasku
- Druh průběhu:**
 - ☐ Standardní funkční modul
 - ☒ Modul schopný remote
 - ☐ Aktualizační modul
 - ☒ Okamž.spušt.
 - ☐ Okamžitý start-ne lze aktualizovat
 - ☐ Opožděný start
 - ☐ Hromad.běh
- Všeobecná data:**
 - Odpovědná osoba: HANA KT
 - Naposled změnil: HANA KT
 - Datum změny: 20.04.2018
 - Paket: ZMG_FW_SOURCE
 - Název programu: SAPLZMG_FWSRC_OBJ
 - Název Include: LZMG_FWSRC_OBJU01
 - Originální jazyk: CS
 - Neuvolněno
 - ☐ Blokování editace
 - ☐ Globální

Obrázek 6.4: Nastavení RFC funkce.

Kromě tohoto nastavení je také zapotřebí nastavit veškeré parametry jako Předání hodnoty. Standardně se totiž předává pouze reference. Z pochopitelných důvodů při vzdáleném spuštění z jiného serveru, či při spuštění na pozadí, není možné používat ukazatele a je nutné předat všechna data hodnotou. Tyto parametry lze nastavit v záložce Import, která je vidět na obrázku 6.4 vedle záložky Vlastnosti. Několikrát bylo zmíněno, že tento funkční modul spouští extrakční funkce na pozadí. Pokud je extrakční funkce nastavena tak, jak bylo právě popsáno, lze ji volat pomocí **CALL FUNCTION** název funkce **IN BACKGROUND TASK**. V tomto případě je pak nutné dát si pozor na to, že při běhu programu je takto volaná funkce pouze naplánována a provedena je až po dosažení implicitně napsané instrukce

COMMIT WORK. Samotné extrakční moduly jsou však již součástí konkrétního využití tohoto frameworku. Pro vytváření extrakčních funkcí však framework obsahuje podporu v podobě sady funkčních modulů, které jsou popsány v kapitole 6.3.4.

6.3.2 Rozložení implementace do paketů

Jedním z vývojových objektů byl objekt paket. Tento vývojový objekt slouží k logickému členění jednotlivých vývojových objektů. Oproti přístupu např. jazyka Javy, zde při potřebě využití vývojového objektu z jiného paketu, než v kterém se aktuálně nacházíme není potřeba žádného klíčového slova `include` apod. Objekty jsou stále globální i přes fakt, že jsou obsaženy v některém z paketů a jedná se tedy doopravdy pouze o logické členění [14].

Název objektu	Popis
<ul style="list-style-type: none"> ▼ ZMG <ul style="list-style-type: none"> ▼ Vložené pakety <ul style="list-style-type: none"> ▼ ZMG_CROSS ▼ ZMG_FW_SOURCE ▼ ZMG_OBJ 	<ul style="list-style-type: none"> Migration - Main MG - Definice dat. typů pro zdrojový i cílový systém MG - Obslužná funkcionality načítání dat Objekty migrace

Název objektu	Popis
<ul style="list-style-type: none"> ▼ ZMG <ul style="list-style-type: none"> ▼ Dítěcí pakety <ul style="list-style-type: none"> ▼ ZMG_CROSS ▼ ZMG_FW_DEST ▼ ZMG_OBJ 	<ul style="list-style-type: none"> Migration - Main MG - Definice dat. typů pro zdrojový i cílový systém MG - Migrační framework na straně cíle Objekty na cíli (například definice struktur migračních obj)

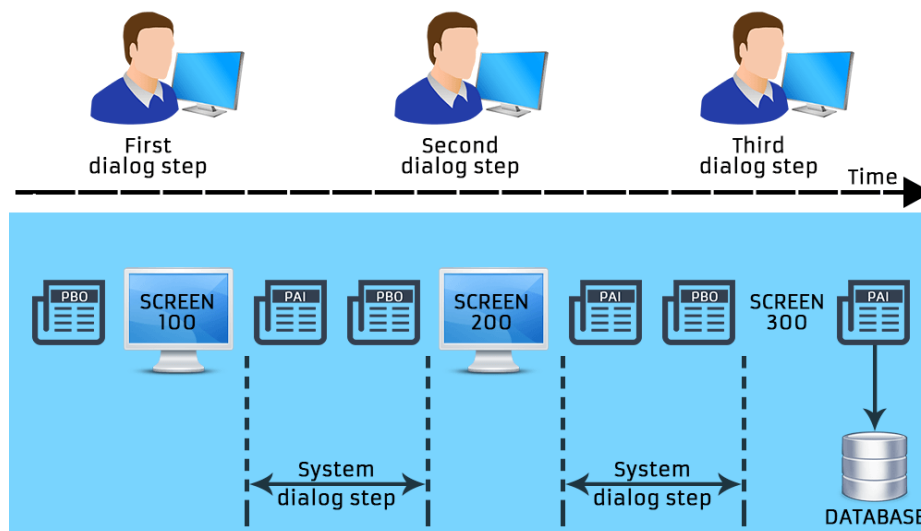
Obrázek 6.5: Hierarchie paketů (zdrojový systém vlevo, cílový vpravo).

Na obrázku 6.5 lze vidět hlavní paket ZMG a jeho verzi jak pro zdrojový, tak cílový systém. Z důvodu RFC funkčních modulů byl založen paket ZMG_CROSS. Jazyk ABAP je silně typově závislý. U funkčních modulů pak tedy musí jednotlivé vstupní a výstupní parametry odpovídat datovým typům, které byly při volání funkce použity. Právě do paketu ZMG_CROSS jsou pak vloženy veškeré definice datových typů, které jsou použity v RFC funkcích a musí tak být na obou systémech zároveň.

Dále pak zahrnuje paket ZMG_FW_SOURCE, resp. ZMG_FW_DEST, který obsahuje implementace migračního frameworku jako takového. V neposlední řadě je zde obsažen paket ZMG_OBJ. Tento paket je určený pro objekty spojené s konkrétní migrací a obecně se bude obsah tohoto paketu lišit při každém použití migračního frameworku na jiných migračních projektech. Na straně zdroje jde většinou o samotné extrakční funkce. Na cílovém systému se pak v prostředí utilitního systému SAP/ISU bude jednat o doprovodné uživatelské implementace pro EMIGALL.

6.3.3 Cílový systém

Cílový systém obsahuje hlavní řídicí program, tedy program pro obsluhu migrace tak, jak byl popsán v podkapitole 5.1. Tento program je klíčovým vizualizačním a koordinačním nástrojem. Pro realizaci tohoto programu byl využit již zmíněný přístup pro tvorbu grafických uživatelských systémů v prostředí ABAP, a to Dynpro. Tento dynamický program pak obsahuje základní oblast pro inicializaci dat, která se jmenuje Report. Na konci této části dynamického programu je pak předáno řízení dále do grafického uživatelského rozhraní. Dynpro pak může obsahovat více obrazovek, kdy pro každou lze definovat tzv. PBO (Process before output) a PAI (Process after input). Jedná se tedy o bloky kódu určené k provedení programové logiky po uživatelské interakci a před opětovným vrácením řízení zpět uživateli. Tento princip je znázorněn na následujícím obrázku 6.6.



Obrázek 6.6: Logika PBO a PAI [1].

Pomocí této logiky byl naprogramován i obslužný program migračního frameworku.

Identifikátor objektu	Typ migračního objektu	Třída	Opakovatelnost	Status	Status popis	Počet objektů	Popis mig. objektu
ACCOUNT	EMIGALL	ZCL_MG_FW_OBJECT_EG			Data načtena	475	SMLUVNÍ ÚČET
ACC_NOTE	EMIGALL	ZCL_MG_FW_OBJECT_EG			Možno spustit	0	POZNÁMKA K SMLUVNÍMU ÚČTU
CONNOBJ	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	460	PŘÍPOJNÝ OBJEKT
DOC_AKONT	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace se nezdařila	143	FICA AKONTACE
DOC_POHL	EMIGALL	ZCL_MG_FW_OBJECT_EG			Data načtena	143	FICA POHLEDÁVKY
DOC_STAT	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	0	FICA STATISTICKÉ POLOŽKY
FACTS	EMIGALL	ZCL_MG_FW_OBJECT_EG			Data načtena	62	ECE FAKTA OM
INSTLN	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	459	ODBĚRNÉ MÍSTO
INSTLNCHA	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	202	ZMĚNA ODBĚRNÉHO MÍSTA
MOVE_IN	EMIGALL	ZCL_MG_FW_OBJECT_EG			Data načtena	458	SMLOUVA - PŘIHLÁŠENÍ
MOVE_OUT	EMIGALL	ZCL_MG_FW_OBJECT_EG			Možno spustit	0	SMLOUVA - ODHLÁŠENÍ
PARTNER	EMIGALL	ZCL_MG_FW_OBJECT_EG			Data načtena	458	OBCHODNÍ PARTNER
PART_REL	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	319	VZTAHY OBCHODNÍHO PARTNERA
POD	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	330	PŘEDÁVACÍ MÍSTO
PODSERVICE	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	330	SLUŽBY PŘEDÁVACÍHO MÍSTA
PREMISE	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace proběhla	321	MÍSTO SPOTŘEBY
TEST1	EMIGALL	ZCL_MG_FW_OBJECT_EG			Možno spustit	0	POPIS
ZBILL_DOC	EMIGALL	ZCL_MG_FW_OBJECT_EG			Migrace se nezdařila	81	ZÚČTOVACÍ DOKLADY PRO M1
ZBIT_EABP	EMIGALL	ZCL_MG_FW_OBJECT_EG			Data načtena	6	ZÁLOHOVÉ PLÁNY EABP
ZCON_NOTE	EMIGALL	ZCL_MG_FW_OBJECT_EG			Možno spustit	0	POZNÁMKA K ODBĚRNÍM OBJEKTŮM

Obrázek 6.7: Obslužný program.

Na obrázku 6.7 je vidět obslužný program naprogramovaný pomocí grafického frameworku Dynpro. Veškeré uživatelské akce, které se týkají standardních grafických prvků, tedy veškerá tlačítka včetně standardních tlačítek SAP GUI v horní liště, je nutné implementovat. Nastavení událostí, které tyto prvky vyvolávají, jsou spravována ve Status GUI. Na Status GUI se dostaneme skrze Workbench tak, jak je vidět na obrázku 6.2. Reakce na tyto akce jsou pak implementovány právě v PAI modulu. Obrazovka také dovoluje dvo-

jitě stisknutí tlačítka myši, a to konkrétně nad tabulkou s objekty. Pokud má objekt stav Data načtena, Migrace se nezdařila, nebo Migrace proběhla, tak je možné pomocí dvojklíku nad popisem stavu, nebo nad názvem objektu zobrazit migrační soubor. Aplikace nemá automatické obnovení a čeká-li uživatel na změnu stavu některého objektu, tak musí pomocí F5, či případně pomocí tlačítka aplikaci obnovovat. Mimo to je zde možnost nastavit si filtr pro migraci, či manuálně vložit binární soubor s daty (tyto možnosti budou vysvětleny v podkapitole 6.4). Dále je zde statistika migrace, kterou lze zobrazit při označení souboru a následném zmáčknutí tlačítka Statistika. Posledním tlačítkem je pak tlačítko Emigall, které je určeno pro přímé přesměrování do nástroje EMIGALL.

6.3.4 Zdrojový systém

Z pohledu frameworku je na straně zdroje implementována sada podpůrných funkčních modulů a také již zmiňovaný migrační log (podkapitola 5.2.2).

Obrázek 6.8: Výběrová obrazovka pro migrační log.

Na obrázku 6.8 je vstupní obrazovka pro zobrazení migračního logu. Skrze tuto obrazovku lze pak přistupovat k již zmíněnému migračnímu logu, který je plněn v průběhu extrahování dat jednotlivých objektů.

Následující obrázek pak ukazuje finální obrazovku pro zobrazení extrakčního logu uživateli:

ID Objektu	Firma	Icon	Důvod nemigrování	Druh Napěťová...	Typ i...	Počet migrovaných	Počet vyřazených	Celkem
INSTLN	Z_NNN	🟢				108.562	0	108.562
INSTLN	Z_NNN	🔴	Jedná se o výkaznické odběrné místo výroby (K_)			0	690	690
INSTLN	Z_NNN	🔴	Jedná se o odběrné místo pro fakturaci lokální spotřeby výrobce			0	1.155	1.155

Obrázek 6.9: Sumarizační log pro objekt INSTLN (odběrné místo).

Byla také zmíněna podpůrná funkcionální, která slouží pro implementaci samotného exportu. Jedná se o tyto funkční moduly:

Z_MG_LOG_SAVE *Vstupní parametry:* tabulka chyb, firma.
Slouží k uložení záznamů o extrakci do migračního logu.

Z_MG_OBJ_SAVE *Vstupní parametry:* adresa cílového systému, ID objektu, firma, samotná data.

Exportuje data, která jsou v předem připraveném formátu do binárního souboru (datová struktura, kterou je potřeba naplnit, je obsažena v migračním frameworku).

Z_MG_OBJ_SAVE_OPEN *Vstupní parametry:* ID objektu, firma.

V případě, že jsou data k exportu načtena pomocí OPEN kurzor a následným FETCH FROM NEXT kurzor, je zapotřebí data postupně přidávat do souboru. Tento modul vytváří soubor pro následné přidávání.

Z_MG_OBJ_SAVE_APPEND *Vstupní parametry:* ID objektu, firma, samotná data.
Přidávání záznamů do souboru.

Z_MG_OBJ_SAVE_CLOSE *Vstupní parametry:* adresa cílového systému, ID objektu, firma.
Konečné uzavření souboru s daty.

Z_MG_OBJ_CREATE_AND_FILL *Vstupní parametry:* datový typ struktury dat, klíč objektu, popis záznamu, data.

Tento funkční modul je užitečný zejména pro EMIGALL migraci. Připravuje data pro následný export do binárního souboru.

Z_MG_OBJ_DOWNLOAD_2_CLIENT *Vstupní parametry:* cesta k souboru, ID objektu.

Funkční modul pro stažení souboru pro migrační objekt ze SAP serveru do lokálního adresáře uživatele.

Paralelizace extrakčních funkcí

Kromě paralelizace spuštěním více extrakčních funkcí zároveň na pozadí, byl v rámci implementace vytvořen vzorový příklad extrakční funkce pro přípojný objekt (CONNOBJ), která využívá princip paralelizace na úrovni načítání dat konkrétního objektu. Obecně jsou data migračního objektu často hledána tak, že je provedeno jedno velké načtení dat z databáze a následně se z mnoha dalších tabulek dohledávají potřebné hodnoty. Byl tedy navržen postup extrakce pomocí paralelizace, kde jsou prvotní data rozdělena do menších bloků. Následně jsou pak paralelně pro tyto menší bloky dohledávány hodnoty. Systém SAP obsahuje mnoho možností paralelizace. Jedna taková efektivní a komplexní možnost se nabízí v podobě SPTA Frameworku [8]. Tento framework je standardní nástroj dodávaný firmou SAP a nachází se ve většině SAP systémů. Princip tohoto frameworku spočívá v implementaci tří povinných bloků. První blok slouží právě k rozdělení vstupních dat. V druhém lze pak naimplementovat vlastní logiku, která bude prováděna paralelně na rozdělených datech. Poslední blok pak slouží k následnému zpracování výsledků z paralelní části.

V případě takovéto paralelní implementace je doporučeno využití funkcí určených pro zápis do souboru `Z_MG_OBJ_SAVE_OPEN`, `Z_MG_OBJ_SAVE_APPEND`, `Z_MG_OBJ_SAVE_CLOSE`. Tyto funkce sloužící ke tvorbě souboru při variantě extrakce pomocí kurzoru, lze jednoduše použít pro paralelní variantu. Před paralelním voláním je soubor otevřen, následně je do něj zapisováno jednotlivými paralelními procesy a nakonec, po ukončení paralelizace, je soubor zavřen.

6.4 Testování výsledného frameworku

Pro toto řešení bylo z firemních důvodů zvoleno klasické uživatelské testování³ na pilotním projektu. Framework byl po dobu dvou měsíců využíván v tzv. testovacích bázích, kdy je stanovený pouze malý vzorek dat, od kterých se odvíjí celá zbylá migrace. Například to může být vzorek 100 zákazníků v systému, k nimž se musí migrovat všechna zbylá potřebná data (domy, přípojky, účty atd.). K podpoře tohoto testování byly vyvinuty některé funkcionality nad rámec základní specifikace, nejprve je však představen princip omezení dat.

6.4.1 Omezení vzorku dat

Jak bylo vidět na obrázku 6.7, obslužný report obsahuje možnost Nastavit filtr. Byla implementována obrazovka, která je použita jako dialogové okno, tzv. pop up a slouží právě ke správě těchto filtrů.

Na obrázku 6.10 lze vidět obrazovku pro správu filtrů. Lze si vytvořit libovolný filtr, který omezuje validní obchodní partnery, smluvní účty, smlouvy, EAN (European Article Number), tedy čárový kód a místo spotřeby. Mimo tyto dané možnosti omezení lze definovat dva další seznamy pro omezení. V praxi je vytvořen jeden filtr pro konkrétní migrační běh, který je pak využíván pro konkrétní migrační testovací běh. Případně lze načít cizí filtr, změnit si potřebné hodnoty a uložit si jej pro konkrétní potřeby. Logika tohoto filtru pak spočívá v tom, že extrakční funkce na straně zdroje dostávají tento filtr na rozhraní a následně díky němu mohou omezit načtení dat ze zdrojové databáze.

³ Jazyk ABAP již umožňuje využití principu Unit testing [14], ale přesto není tento princip pořád hojně využíván.

Obrázek 6.10: Nastavení filtru.

6.4.2 Další implementace spojené s testovacími běhy

V systému SAP často bývá problém s oprávněními a nastaveními samotného systému, kdy nastavení z důvodu administrativních překážek trvá dlouho. Některá nastavení mohou provádět pouze zaměstnanci firmy SAP, do celého procesu tak mimo migrační tým a zákazníka vstupuje ještě třetí strana v podobě firmy SAP, či dokonce jiné společnosti, která spravuje přístupy apod. Mezi hlavní problémy při realizaci migrace pomocí migračního frameworku patří:

- chybějící spojení mezi systémy,
- konkrétní RFC funkce musí být povoleny administrátorem,
- chybějící sdílené úložiště pro migrační soubory.

Byly tedy navrženy funkcionality, které simulují, či řekněme zastupují a dočasně řeší tyto problémy.

Spuštění extrakce bez RFC spojení (v případě migrace pomocí souborů)

Vzdálené spouštění extrakčních funkcí bez nutnosti být připojen na zdrojový systém, je jednou ze základních funkcností. Bohužel kvůli výše zmíněným problémům může tento problém nastat velice často. Pro tyto případy byl tedy naimplementován program pro spouštění ZMG_FWSRC_OBJ_EXPORT_RUN, kterému odpovídá transakce ZMGEXP. Tento program obsahuje veškerou funkcionalitu nutnou k nasimulování vzdáleného volání včetně možnosti zadat filtr migrace. Zde není potřeba tvořit vlastní logiku uložení filtru, neboť SAP GUI umožňuje vytvářet tzv. varianty spuštění programu. V případě, že má program nějaká vstupní pole, je možné si uložit variantu spuštění tohoto programu a tu používat opakovaně.

Stažení exportovaného souboru do lokálního úložiště

Implementace ukládacích funkčních modulů má v sobě vestavěnou funkcionalitu na automatické nabídnutí stažení právě exportovaného souboru. SAP má již vestavěné trans-

akce na stažení/nahrání souboru ze/do SAP serveru⁴. Je tedy naimplementován program, který značně zjednodušuje tento proces. Jedná se o program ZMG_FILE_DOWNLOAD_2_LOCAL, ke kterému byla vytvořena transakce ZMGDWN.

6.4.3 Vyhodnocení

Migrační framework byl testován desítkou migračních techniků při realizaci migračních testovacích běhů. Testování proběhlo pouze uživatelsky a ověřovalo předem danou specifikaci tak, jak byla popsána v kapitole 5. Mimo to byl při vývoji použit ABAP Test Cockpit [12, 16]. Tento nástroj umožňuje provádět automatickou analýzu zdrojového kódu a to například na úrovni optimalizace jednotlivých databázových příkazů. Dále provádí analýzu kódu na nedosažitelné bloky kódu apod.

⁴ CG3Y a CG3Z, jejichž použití pro časté stahování souborů z konkrétního adresáře je však poněkud zdržující.

Kapitola 7

Závěr

Cílem této diplomové práce bylo analyzovat možnosti migrace dat mezi SAP systémy, konkrétně pak se zaměřením na odvětvové řešení SAP/ISU, které je šité na míru firmám působícím na trhu s elektřinou, plynem atd. Na základě této analýzy a požadavků firmy CGI IT Czech Republic s. r. o. bylo cílem následně navrhnout a implementovat migrační framework pro podporu migrace mezi systémy SAP/ISU s jeho rozšířením SAP/CRM. Práce je členěna do dvou logických celků.

První část práce zhodnotila problematiku SAP řešení, a to především z pohledu architektury těchto systémů se zaměřením na datový model a možnosti zákaznického vývoje v SAP systémech 2. V kapitole 3 byly analyzovány různé druhy možností, jak přenést data mezi jednotlivými systémy, tedy jak je možné data migrovat do nového systému. V rámci této části byl v kapitole 4 představen i konkrétní motivační příklad zdroje potřeby provádět takovouto migraci dat, tzv. oddělení tržních celků společností (též označováno jako Unbundling) zabývajících se výrobou, prodejem a distribucí elektřiny. Přesně takovéto společnosti hojně využívají produkt SAP/ISU. Na tomto konkrétně daném příkladu bylo provedeno logické členění dat určených pro migraci a byly vymezeny požadavky na systémy, které budou v rámci této migrace použity.

Na základě poznatků z právě popsané první části této práce a firemních požadavků, byl definován návrh migračního frameworku. Tento návrh je detailně popsán v kapitole 5, která se zabývá jednotlivými návrhy pro zdrojový a cílový systém. V neposlední řadě byl v kapitole 6 popsán způsob implementace tohoto frameworku v jazyce ABAP. Tato část popisuje i způsob, jakým byla ověřena funkčnost tohoto frameworku.

Povedlo se tedy analyzovat problematiku migrací v prostředí SAP a následně implementovat framework pro podporu migrace v odvětvovém systému SAP/ISU. Jako rozšíření do budoucna je počítáno s využitím paralelizace. Extrakce na straně zdrojového systému, ale také finální uložení dat do nového systému, jsou při testovacích migračních bězích časově únosné, ale při objemu dat při finální migraci by tento výkon byl nedostačující. Proto byl nastíněn také způsob paralelizace extrakčních funkcí. Dalším potřebným rozšířením může být komprimace exportovaných souborů, neboť v aktuálním řešení jsou soubory kódovány pomocí UTF-8 a dosahují poměrně značné velikosti.

Literatura

- [1] Dialog Programming Tutorial: Module Pool in SAP ABAP. [Online; navštíveno 23.4.2018].
URL <https://www.guru99.com/dialog-programming-tutorial.html>
- [2] Dynpro ABAP. [Online; navštíveno 29.4.2018].
URL https://eaexplorer.hana.ondemand.com/_item.html?id=10511#!/overview
- [3] IS-U / CRM Integration – Data Model. [Online; navštíveno 28.4.2018].
URL <https://help.sap.com/viewer/af9bfbef2af4c31aaad23efbd40b9c5/6.04.19/en-US/3c81ce53118d4308e1000000a174cb4.html>
- [4] Migration - Utilities Industry. [Online; navštíveno 29.4.2018].
URL <https://wiki.scn.sap.com/wiki/display/UIndustry/Migration>
- [5] Position of the Basis System Within the R/3 System. [Online; navštíveno 28.4.2018].
URL https://help.sap.com/doc/saphelp_47x200/4.7.2/en-US/fc/eb2e97358411d1829f0000e829fbfe/content.htm?no_cache=true
- [6] SET HANDLER - FOR - ABAP Documentation. [Online; navštíveno 29.4.2018].
URL https://help.sap.com/doc/abapdocu_750_index_htm/7.50/en-US/abapset_handler_instance.htm
- [7] Shared Objects - Implementation. [Online; navštíveno 29.4.2018].
URL https://help.sap.com/saphelp_nw70/helpdata/en/c5/85634e53d422409f0975aa9a551297/frameset.htm
- [8] SPTA Framework - Parallel Processing Technique in SAP ABAP. [Online; navštíveno 29.4.2018].
URL <http://mysapbook.blogspot.cz/2013/12/spta-framework-parallel-processing.html>
- [9] IS-U SAP Utilities SAP ERP, Central Component Release 6.0, Release Notes. SAP AG, 2006, [Online; navštíveno 23.12.2017].
URL [https://help.sap.com/http.svc/rc/acd5ba39aadb4678a6d4c0424a5551ec/6.0.0/en-US/Chapter_40__IS-U_SAP_UilitiesE_\(2\).PDF](https://help.sap.com/http.svc/rc/acd5ba39aadb4678a6d4c0424a5551ec/6.0.0/en-US/Chapter_40__IS-U_SAP_UilitiesE_(2).PDF)
- [10] *Performance of the Initial Download from SAP for Utilities into SAP CRM*. Walldorf: SAP AG, 2010.
- [11] Anderson, G. W.: *Naučte se SAP za 24 hodin*. Brno: Computer Press, první vydání, 2012, ISBN 978-80-251-3685-0.

- [12] Hardy, P.: *ABAP to the Future*. Rheinwerk Publishing, 2016, ISBN 978-1-4932-1411-2.
- [13] Harmuth, T.: ABAP in Eclipse jetzt mit SAP NetWeaver AS ABAP 7.03/7.31 SP4 verfügbar. itelligence, [Online; navštíveno 22.04.2018].
URL <https://itelligencegroup.com/de/local-blog/abap-in-eclipse-jetzt-mit-sap-netweaver-as-abap-7-037-31-sp4-verfuegbar/>
- [14] Heilman, R.; Jung, T.: *Next Generation ABAP Development*. Galileo Press, 2007, ISBN 978-1-59229-139-7.
- [15] Hernández, J. A.; Keogh, J.; Martinez, F. F.: *SAP R/3 Handbook*. Rheinwerk Publishing, Third Edition, 2006, ISBN 0-07-225716-4.
- [16] Hildenbrand, P.: ABAP Test Cockpit. [Online; navštíveno 29.4.2018].
URL <https://wiki.scn.sap.com/wiki/display/ABAP/ABAP+Test+Cockpit>
- [17] Jadhav, S.: SAP IS-U Migration Workbench: Step by Step EMIGALL . [Online; navštíveno 4.5.2018].
URL <https://archive.sap.com/kmuuid2/70061527-8ef5-2d10-91bb-f957769a402b/SAP%20IS-U%20Migration%20Workbench%3A%20Step%20by%20Step%20EMIGALL.pdf>
- [18] Katz, E.: Legacy System Migration Workbench. [Online; navštíveno 28.4.2018].
URL <https://archive.sap.com/documents/docs/DOC-26159>
- [19] Kühnhauser, K.-H.: *ABAP Výukový kurz*. Brno: Computer Press, 2009, ISBN 978-80-251-2117-7.
- [20] Maassen, A.; Schoenen, M.; Frick, D.; aj.: *SAP R/3: Kompletní průvodce*. Brno: Computer Press, 2007, ISBN 978-80-251-1750-7.
- [21] Rybička, J.: *Zákaznický vývoj v systému SAP*. Praha: Vysoká škola ekonomická v Praze, 2010.
- [22] Vondráček, M.; Skuček, T.: *Náklady unbundlingu v energetice: konkurenční prostředí ve prospěch zákazníka, nebo příliš drahý experiment?* Praha: Vysoká škola ekonomická v Praze, 2010.
URL <https://www.vse.cz/polek/download.php?jnl=eam&pdf=102.pdf>
- [23] Willinger, M.; Gradl, J.: *Migrating your SAP data*. Bonn: Galileo Press, 2008, ISBN 978-1592291700.
- [24] Wood, J.: *ABAP cookbook: programming recipes for everyday solutions*. Galileo Press, 2010, ISBN 978-1-59229-326-1.

Příloha A

Obash CD

- /Framework/Source_system/ZMG/ – Zdrojové soubory Frameworku na straně zdrojového systému
 - Datové typy¹ ZMG_CROSS/
 - Zdrojové soubory migračního frameworku ZMG_FW_DEST/
 - Ukázkové implementace exportních funkcí ZMG_OBJ_DEST/
- /Framework/Target_system/ZMG/ – Zdrojové soubory Frameworku na straně cílového systému
 - Datové typy (opět pro názornost) ZMG_CROSS/
 - Zdrojové soubory migračního frameworku ZMG_FW_DEST/
 - Místo pro dodatečné zákaznické implementace ZMG_OBJ_DEST/
- /thesis/ – Technická zpráva ve formátu PDF v adresáři
- /thesis_latex/ – Zdrojové soubory technické zprávy v adresáři

Implementace ze systému SAP nelze obecně exportovat a jedná se tedy v případě zdrojových souborů o veškeré implementace, ke kterým se lze skrze SAP GUI dostat. V případě přenesení těchto souborů je přesto nutné založit odpovídající objekty v repositáři vývojových objektů a provést jednotlivá nastavení tak, jak bylo popsáno v jednotlivých kapitolách.

¹ Zde bohužel jen pro názornost, neboť typy nelze ze SAP systému exportovat a jsou uloženy v Object Repository.